

אלגוריתמים 1 - סיכום הרצאות לבחן

27 בינואר 2026

הסיכום נכתב תוך כדי הרצאות סמס א' תשפ"ו (2026) ולכן יתכן שנפלו טעויות במהלך כתיבת הסיכום, ככה שעל אחריותכם.
גיא ערד-און.

תוכן עניינים

4	אלגוריתם קרטסובה	1
5	הרצאה 1 - <i>FFT</i>	2
5	פעולות של פולינומיים	2.1
6	יצוג של פולינומיים	2.2
6	יצוג ע"י מקדמים	2.2.1
6	יצוג ע"י נקודות	2.2.2
9	סיכום הפעולות בשיטות השונות	2.2.3
9	אלגוריתם לכפל פולינומיים מהיר: התמרת פורייה <i>FFT</i>	2.3
10	תכונת הנגדיות החלשה	2.3.1
11	תכונת הנגדיות חזקה	2.3.2
11	איזה מספרים מקיימים את תכונת הנגדיות חזקה?	2.3.3
12	האלגוריתם <i>FFT</i>	2.4
13	כיצד נעבורCut משיטות הנקודות חזקה לשיטת המקדמים?	2.5
14	תרגילים <i>FFT</i>	2.6
14	כפל פולינומיים מדרגות חסומות שונות	2.6.1
15	בעיית <i>3SUM</i>	2.6.2
15	בעיית חישוב מרחק האמיניג	2.6.3
18	הרצאה 2 - <i>MST</i>	3
18	ע"ז פורש מינימום	3.1
19	בעיית מציאת ע"ז פורש מינימום	3.2
19	אלגוריתם חמדניים (<i>Greedy</i>)	3.3
19	למה הבחירה החמדנית	3.4
21	כיווץ קשותות	3.5
22	אלגוריתם גנרי של <i>MST</i>	3.6
23	תכונת תת המבנה האופטימלי	3.7
24	האלגוריתם של פרים (<i>Prim</i>)	3.8
26	האלגוריתם של קروسקל	3.9
27	תכונת המעלגים הכבדים של <i>MST</i> (תרגול)	3.10

28	השפעת סדר מין הקשתות על הפלט בהרצאת האלגוריתם של קורוסקל	3.11	
30	הרצאות 3 + 4 - <i>shorts path – SSSP</i>	4	4
30	בעיית מציאת המסלול הקצר ביותר	4.1	
31	איך נראה בכל אחד מהגרסאות כמשמעותם את המסלול?	4.1.1	
32	אלגוריתם <i>SSSP – BFS</i> במקרה הלא ממושקל	4.2	
33	האלגוריתם $BFS = (G = (V, E), s)$	4.2.1	
34	נכונות של <i>BFS</i>	4.2.2	
35	אלגוריתם סריקת <i>DFS</i>	4.3	
37	סיווג קשתות	4.3.1	
37	גרף מכון חסר מעגלים (<i>DAG</i>)	4.4	
37	מיון טופולוגי	4.5	
38	רכיבים קשורים היבט (G^{SCC})	4.6	
40	מציאת גרף דו צדדי	4.7	
40	מציאת עץ מסלולים קצרים ביותר בגמ"ל	4.8	
41	<i>SSSP</i> בגרפים ממושקלים	4.9	
41	נסיין ראשון - תכונות דינמי	4.9.1	
42	סוגי מעגלים	4.9.2	
42	הקלת קשתות – <i>Relaxtions</i>	4.9.3	
44	אלגוריתם מבוסס הקלות והוכחת נכונות של בלמן פורד	4.9.4	
45	האלגוריתם של בלמן פורד	4.9.5	
46	האלגוריתם של דיקסטרה	4.10	
46	הגדרת הבעה ומבוא	4.10.1	
47	האלגוריתם	4.10.2	
48	הוכחת נכונות של דיקסטרה	4.10.3	
49	סיכום	4.11	
49	הרצאה 5 : <i>shorts path – APSP</i>	5	
49	מבוא לכפל מטריצות מהיר	5.1	
50	הגדרת <i>APSP</i>	5.2	
50	האלגוריתם של <i>Floyd – Warshall</i>	5.3	
51	הסגור הטרניזיטיבי של גרף מכון	5.4	
52	חישוב הסגור הטרניזיטיבי בזמן $O(V ^{\omega})$	5.4.1	
53	האלגוריתם של <i>Jhonson</i>	5.5	
55	האלגוריתם של <i>Seidel</i>	5.6	
55	הגדרת הבעה	5.6.1	
55	כפל מטריצות בוליאני	5.6.2	
56	טענות 1, 2	5.6.3	
59	האלגוריתם	5.6.4	
61	A^*	5.7	
64	הגדרה פורמלית	5.7.1	
65	הרצאה 6 : רשתות זרימה (<i>Network Flow</i>)	6	
65	הגדרה פורמלית של זרימה	6.1	
67	הגדרת הבעה	6.2	
67	תכונות של זרימה	6.3	
70	שיטת פורד-פלקרטון	6.4	
71	הרשת השיוורית	6.5	
71	שיטת פורד-פלקרטון	6.6	

72	ncונות האלגוריתם וזמן הריצה	6.7
73	משפט $max-flow - min-cut$	6.7.1
74	סיבוכיות זמן הריצה (פורד פרקלסן)	6.7.2
74	מציאת חתך (s, t) מינימום ברשת זרימה	6.8
75	הכרעה האם זרימת מקסימום \ חתך מינימום ייחודיים	6.9
76	הרצאה 7 : זרימה - אדמונס קארפ <i>Hopcroft - Karp</i>	7
76	האלגוריתם של אדמונס קארפ	7.1
79	גרף השכבות	7.2
80	מציאת מסלול	7.3
81	עדכון גרף השכבות	7.4
81	האלגוריתם של <i>Dinic</i>	7.5
83	האלגוריתם של <i>Hopcroft - Karp</i>	7.6
86	זיווג מקסימום בגרף דו צדי	7.7
87	זרימה אי זוגית	7.8
88	הרצאה 8 : <i>BMM</i> , <i>BMM</i> מושלמים וכל כבד	8
88	<i>BMM</i>	8.1
89	<i>BMM</i> קומבינטוריה והשערת <i>BMM</i>	8.1.1
90	זיהוי מושלמים בגרף	8.2
93	TD (מציאת מושלמים) בגרפים דילילים (כל כבד)	8.3
94	בעיית דיווח המושלמים (כל כבד)	8.4
96	מרקח האמינג עבור א"ב כללי (כל כבד)	8.5
98	הרצאה 9 : אלגוריתמים רנדומיים	9
98	מבוא והגדלה	9.1
99	9.1.1 אלגוריתמי מונטה קרלו ואלגוריתמי לאס וגאס	
99	VIDOA כפל מטריצות	9.2
99	נסיון ראשון	9.2.1
101	נסיון שני - Amplification (Amplification)	9.2.2
102	מיון מהיר - <i>Quick Sort</i> : מבוא	9.3
103	מיון מהיר פרנוואידי	9.4
104	תוחולת זמן הריצה של מיון מהיר "קלאסי"	9.5
105	מיון דלי (Bucket Sort)	9.6
106	הרצאה 10 : שבירת סימטריה	10
106	מבוא והגדלת הבעיה	10.1
108	המודל המבוזר המקומי	10.2
109	בחירה <i>ID</i> במודל המפואר	10.3
109	בעיית צביעה	10.4
110	צביעה במודל המבוזר המקומי	10.5
111	אלגוריתם צביעה	10.6
111	10.6.1 ncונות האלגוריתם	
114	בעיית אוסף הקופונים (איסוף מדבקות לאלבום - תרגול)	10.7
114	10.7.1 תוחלת מספר הרטיטיסים שיש להוציא עד הוצאה כל הסוגים	
114	חסם עליון על מס' הקלפים שנצטרכז להוציא בהסתברות גבואה	10.7.2
114	קבוצה פוגעת (Hitting - Set)	10.8
115	ערובב אחד (תרגול)	10.9

	שיטת ראיונה: הפלת הדפים אל הרצפה - ואיסופם מימין	10.9.1
116	לשמאל	
	שיטת שנייה: "הווצאת האיברים מתוך סל זה אחריו זה,	10.9.2
117	כשכל הווצה נעשה נושא באקראי בהתפלגות אחידה" . . .	
118	118 Quick Select ו Skip List (תרגול מס' 11)	11
118	תיאור המבנה	11.1
118	ניתוח סיבוכיות מקום וזמן ריצה של המבנה	11.2
119	זמן פולחת חיפוש	11.3
120	Quick Select	11.4
122	הרצתה 11: חסמי Chernoff	12
122	אי שוויון מוקוב וצ'בישב	12.1
124	חסמי צ'רנוף: הגדרה	12.2
125	הוכחת אי שוויון צ'רנוף	12.3
126	מיון מהיר (Quick Sort)	12.4
128	ניסוח שונה לאי שוויון צ'רנוף	12.5
128	הרצתה 12: MST מתקדם	13
128	מבוא	13.1
128	Boruvka	13.2
129	שייפור	13.3
131	האלגוריתם של (KKT) Karger – Klein – Tarjan	13.4
133	פסאודו של KJ	13.5
134	זמן הריצה	13.6
135	סיכום אלגוריתמים שראינו בקורס + זמן ריצה (קופסאות שחורות")	14

1 אלגוריתם קרטסובה

הבעיה: נתונות שני מחרוזות בעלות n ביטים כל אחת. נרצה לכפול את המחרוזות. מה סיבוכיות הפעולה?

פתרון: מכפילים ביטים. יש n^2 מכפלות כאלה. נחשב על הפתרון באמצעות רקורסיה. ככלומר, במקומות להכפיל מחרוזת באורך n נרצה לצלצם את הכפל לשמשו כמו $\frac{n}{2}$. בהינתן מס' x נרצה לרשום אותו אחרת. נחלקו לשני חלקים ונקבל כי $X = x_1 * 2^{\frac{n}{2}} + x_2$. כאשר נכפיל ב- $2^{\frac{n}{2}}$ זה לא באמת עולה לי כי אנחנו רק מזיזים את הביטים. (כלומר – בהינתן 1234. נוכל לרשום כי $1234 = 12 * 10^2 + 34$)
 – ככלומר הכפל אכן לא באמת עולה
 – כתע יש מס' נוסף – $Y = y_1 * 2^{\frac{n}{2}} + y_2$
 נראה כי אם נכפול נקבל –

$$xy = (x_1 * 2^{\frac{n}{2}} + x_2)(y_1 * 2^{\frac{n}{2}} + y_2) = x_1y_12^n + x_1y_22^{\frac{n}{2}} + x_2y_12^{\frac{n}{2}} + x_2y_2$$

מה קיבלנו כאן? נראה כי כל הפעולות של 2 בחקת ה- n פעולות האזת ביטים ולמעשה כאשר נכפיל שני מחרוזות צמצמנו את הבעיה שלנו ל- 4 – תתי בעיות בגודל $\frac{n}{2}$!
 מכאן נקבל כי נוסחת הנסיגה היא –

$$T(n) = 4T\left(\frac{n}{2}\right) + O(n) = \Theta(n^2)$$

זה לא טוב - מדוע? זה בדיק כמו הפתרון הנאי. בואו ננסה לצמצם את מס' הקריאה
הרכורסיביות:
נגיד "

$$A = x_1 y_1, B = x_2 y_2$$

זכיר שנרצה לחשב את

$$C = (x_1 + x_2)(y_1 + y_2) = x_1 y_1 + x_1 y_2 + x_2 y_1 + x_2 y_2$$

cut נראה כי

$$xy = x_1 y_1 2^n + x_1 y_2 2^{\frac{n}{2}} + x_2 y_1 2^{\frac{n}{2}} + x_2 y_2 = x_1 y_1 2^n + 2^{\frac{n}{2}}(x_1 y_2 + x_2 y_1) + x_2 y_2 = x_1 y_1 2^n + 2^{\frac{n}{2}}(C - B - A) + x_2 y_2$$

מכאן שקיבלו נוסחה עם 3 איברים cut (יש אישרו קבוע באיבר האמצע)! ולא 4 כמו קודם,
נוכל לרשום -

$$T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n)$$

$$. T(n) = \Theta(n^{\log 3}) \approx \Theta(n^{1.58})$$

2 הרצאה 1 - 2

2.1 פעולות של פולינומים

פולינום ניתן לכתיבה כך - $P(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{n-1} x^{n-1} = \sum_{i=0}^{n-1} a_i x^i$
פולינום זה הוא מדרגה n , והוא מדרגה חסומה k לכל $n+1, n, k \geq n$. (כלומר $P(x)$ חסום $n+2, n+1, n, n+2$ וכו').

1. **חיבור/חיסור פולינומים:** יהיו $A(x) = \sum_{i=0}^{n-1} a_i x^i, B(x) = \sum_{i=0}^{n-1} b_i x^i$ פולינומים. נגיד
את החיבור/חיסור של $A(x)$ ו- $B(x)$

$$C(x) = A(x) \pm B(x) = \sum_{i=0}^{n-1} (a_i \pm b_i) x^i$$

נשים לב כי דרגת הפולינום $C(x)$ נשארה זהה לדרגה של $A(x), B(x)$.

2. **כפל פולינומים:** יהיו $A(x) = \sum_{i=0}^{n-1} a_i x^i, B(x) = \sum_{i=0}^{n-1} b_i x^i$ פולינומים. נגיד את הכפל
של $B(x)$ ו- $A(x)$

$$C(x) = A(x) \times B(x) = \sum_{i=0}^{2n-2} c_i x^i$$

בasher zo koraiim kovaloachya.
 nshim lab ci drat ha-polinom C tihha
 $.2(n-1) = 2n - 2$

3. **חישוב ערך:** יהי פולינום $A(x) = \sum_{i=0}^{n-1} a_i x^i$, בהינתן ערך x_0 , נרצה לחשב את $A(x_0)$.

2.2 יציג של פולינומים

יהי פולינום $A(x) = \sum_{i=0}^{n-1} a_i x^i$. נרצה להראות מספר דרכים ליצג את הפולינום:

2.2.1 יציג ע"י מקדמים

נרצה לשמור את המקדמים בלבד של הפולינום. נשתמש במערך ARR בגודל n , ונשמר בתוכו את המקדמים:

$$ARR = (a_0, a_1, \dots, a_n)$$

"ציג זה באמצעות מקדמים נוחש לטוב, קיימת פונקציה $ch(x)$ ו

לעולם ה"ציגים" לעולם הפולינומים" - מה הכוונה? לא ניתן שנקבל "ציג זהה עבור $A_2(x) \neq A_1(x)$ ולא ניתן "ציג שונה עבור $A_1(x) = A_2(x)$.
חיבור/חיסור: יהי פולינומים $A(x), B(x)$ חסומים מדרגה $1-n$, אז נרצה לחשב את $C(x) = A(x) \pm B(x)$

$$\forall_{0 \leq i \leq n-1} : c_i = a_i \pm b_i$$

נשים lab ci bahinten shirot ha-makdimim, lechab polinom (C(x) hanil yulah) O(n) u'i chibor/chisor zoog ha-uracim bataim A[i], B[i]. lamza achd ha-muracim. lema'aseh gom la natzar shiush b'makom nosaf. seh"b - chibor/chisor (O(n)).
כפל: lepi hanoscha shatorda leili:

$$\forall_{0 \leq i \leq n-1} c_i = \sum_{j=0}^i a_j b_{i-j}$$

נראה ci zman lechab makdm c_i bood yulah (O(n) zman, vlen chab kol ha-makdimim, kolmor **חישוב הכפל יعلاה** O(n²) zman).

חישוב ערך: בהינתן x_0 נרצה לחשב את $A(x_0)$. לא יהיה כאן רעיון מתחכם - נחשב את $x_0^0, x_0^1, \dots, x_0^{n-1}$ ולאחר מכן נחשב את החיבור $a_0x_0^0 + a_1x_0^1 + \dots + a_{n-1}x_0^{n-1}$ וסה"כ **חישוב ערך** $O(n)$.

2.2.2 יציג ע"י נקודות

bahinten yishr makkbil laachd ha-zirrim, baamutzut nukoda achot nitun ledut latzar at ha-yishr. bahinten shiduoy ci ha-yishr hoa ko linari yishr = mx + b = y ai bahinten shati nukodot nitun ledut baofen madik at meshovat ha-yishr, bahinten farbola y = ax² + bx + c = y bahinten shlosh nukodot nitun ledut baofen madik at meshovat ha-yishr vkn halala. baofen calli, ya polinom $A(x) = \sum_{i=0}^{n-1} a_i x^i$ ai baamutzut n

נקודות $(x_0, x_1, \dots, x_{n-1})$ ניתן לדעת באופן מדויק ולתאר את הפולינום, **בתנאי שהנקודות שונות אחת מהשנייה.**
סה"כ נציג את הפולינום A באמצעות הנקודות:

$$(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})$$

באשר $\forall 0 \leq i \leq n-1 : y_i = A(x_i)$.

האם הייצוג הזה טוב? האמת שכן, כיצד נראה זאת? צריך להראות שהייצוג הוא חח"ע ועל. כמובן, שקיימת פונקציה חח"ע ועל בין הייצוג לפולינומים.
נשים לב כי $y_i = A(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + \dots + a_{n-1} x_i^{n-1}$.
נראה כי ניתן לקבל בסה"כ n משוואות, ולפתור מערכת של n -משוואות, ומצאנו כך את כל ערכי $a_0, a_1, a_2, \dots, a_n$. ולכן סה"כ הייצוג הוא חח"ע ועל. באופן פורמלי יותר - נשים לב כי מערכת המשוואות הנ"ל ניתנת לתיאור בצורה מטריציתנית:

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} x_0^0 & x_0^1 & x_0^2 & \dots & x_0^{n-1} \\ x_1^0 & x_1^1 & x_1^2 & \dots & x_1^{n-1} \\ x_2^0 & x_2^1 & x_2^2 & \dots & x_2^{n-1} \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ x_{n-1}^0 & x_{n-1}^1 & x_{n-1}^2 & \dots & x_{n-1}^{n-1} \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

מטריצה זו נקראת **מטריצת וNDERMONDAH**. נקרא למטריצה V , לוקטור הימני נקרא \vec{a} ולוקטור השמאלי נקרא \vec{b} .
טענה: אם במטריצת וNDERMONDAH ערכי $(x_0, x_1, \dots, x_{n-1})$ שונים זה מזה, אז מטריצת וNDERMONDAH היפה.

כיון ש V היפה אצלו, נשים לב כי $\vec{b} = V^{-1} \vec{a}$. סה"כ מצאנו דרך לעبور בין ערכי הנקודות ולקבל את המקבדים של הפולינום, ולכן ייצוג זה הוא חח"ע ועל.

كيف עבורים בין ייצוג עי מקדמים לייצוג עי נקודות? עי n פעולות של חישוב ערך. נבחר $x_{n-1} \neq x_{n-2} \neq \dots \neq x_1 \neq x_0$ ונחשב $A(x_0), \dots, A(x_{n-1})$. כמה עולה מעבר זה? כל חישוב עולה $O(n)$ ולכן סה"כ n חישובים עולה לנו $O(n^2)$.
كيف עבורים בין ייצוג עי נקודות לייצוג עי מקדמים? פעולה זו יש שם - אינטראפולציה. משתמשים בנוסחת לגראנז:

$$A(x) = \sum_{i=0}^{n-1} y_i \cdot \frac{\prod_{j \neq i} (x - x_j)}{\prod_{j \neq i} (x_i - x_j)}$$

נשים לב כי חישוב זה עולה $O(n^2)$ זמן, וכך גם המעבר השני עולה כמו המעבר הראשון.

1. **חיבור:** יהיו שני פולינומים:
א. **נניח כי שני הפולינומים מוצגים עי אותן נקודות** $(x_0, x_1, \dots, x_{n-1})$

$$A = (x_0, A(x_0)), (x_1, A(x_1)), \dots, (x_{n-1}, A(x_{n-1}))$$

$$B = (x_0, B(x_0)), (x_1, B(x_1)), \dots, (x_{n-1}, B(x_{n-1}))$$

נשים לב כי במקרה זה נציג את פולינום החיבור:

$$C = (x_0, C(x_0)), (x_1, C(x_1)), \dots, (x_{n-1}, C(x_{n-1}))$$

באשר $O(n)$ זם. $\forall i \leq n-1$ $C(x_i) = A(x_i) + B(x_i)$. נשים לב כי בדרך זו, חיבור פולינומים עולה $O(n^2)$.

ב. שני הפולינומים לא בהכרח מיוצגים ע"י אותן נקודות.
אין דרך קסם. מה שנעשה יהיה לבצע אינטראפלציה, באמצעות נוסחת לגראנץ. נדרש ליצוג ע"י מקדמים של שני הפולינומים, זה יעלה עבור כל פולינום $O(n^2)$. אchar'כ נחבר את שני הפולינומים בשיטת המקדמים, מה שיעלה עוד $O(n)$, ואchar'כ נמיר חזקה את פולינום החיבור משייטת המקדמים חזקה לשיטת הנקודות, מה שיעלה עוד $O(n^2)$. סה"כ - $O(n^2)$ לחיבור פולינומים.

2. כפל: יהיו שני פולינומים.

א. נניח כי שני הפולינומים מיוצגים ע"י אותן נקודות $(x_0, x_1, \dots, x_{2n-1})$

$$A = (x_0, A(x_0)), (x_1, A(x_1)), \dots, (x_{n-1}, A(x_{n-1}))$$

$$B = (x_0, B(x_0)), (x_1, B(x_1)), \dots, (x_{n-1}, B(x_{n-1}))$$

נשים לב כי במקרה זה נציג את פולינום הכפל:

$$C = (x_0, C(x_0)), (x_1, C(x_1)), \dots, (x_{n-1}, C(x_{n-1}))$$

באשר $O(n)$ זם. $\forall i \leq 2n-1$ $C(x_i) = A(x_i) \times B(x_i)$. נשים לב כי בדרך זו, כפל פולינומים עולה $O(n^2)$.

נשים לב כי - הדרגה של פולינום הכפל C היא $2n-2$. ככלمر צריך לציג אותו באמצעות $2n-2$ ארכים. לכן בנויגוד למקרים אחרים, כאן דרשו מ- A ו- B להיות מיוצגים ע"י $2n-2$ נקודות. אחרת, לא יוכל לכפול.

ב. שני הפולינומים לא בהכרח מיוצגים ע"י אותן נקודות.

באופן דומה, אין פתרון קסם. לבצע אינטראפלציה. נדרש לשיטת המקדמים, שם נכפול (n^2) . אchar'כ נשתמש חזקה באינטראפלציה לעבור חזקה לשיטת הנקודות. סה"כ $O(n^2)$ לכפל פולינומים במקרה זה.

3. חישוב ערך:

בכל מקרה, צריך לבצע אינטראפלציה אז לחשב ולכן $O(n^2)$.
הערה: אם אנחנו במקרה בו ערכי x של שני הפולינומים זהים, והערך שנקראנו לחשב הוא כבר אחד מהערכים שאיתם קיבלנו את הפולינום, כל שיש לעשות בשביל לחשב את הערך הוא לחפש את ערך האיקס הספציפי. מה שיעלה לנו $O(log n)$ בהנחה שהנקודות מסודרות בסדר עולה ביחס לערך האיקס.

2.2.3 סיכום הפעולות בשיטות השונות

סוג השיטה / פעולות	מקדמים	שיטת הנקודות - אוטם ערכיא	שיטת הנקודות - לא בהכרח אוטם ערכיא
חיבור/חיסור	$O(n)$	$O(n)$	$O(n^2)$
כפל	$O(n^2)$	($O(n^2)$: ציריך שיהו 2^{n-2} ערכים שונים לכל פולינום בשביל שמכל לכפול.)	$O(n^2)$
חישוב ערך	$O(n)$	$O(2^n)$	$O(n^2)$

2.3 אלגוריתם לכפל פולינומים מהיר: התמרת פורייה FFT

יהיו A, B פולינומים המוגרים ע"י מקדמים. נרצה לקבל את $C = A \times B$. ראיינו שאפשר ב($O(n^2)$ בקורס מבני נתוניים), ראיינו דרך מעניינת לכפל מטריצות (דומה לפולינומים באמצעות הפרד ומשולב ב($O(n \log n)$).Cut נרצה למצוא שיטה ב($O(n \log n)$).

מה ראיינו עד כה? קיבל מקדמים. תבצע $2n - 2$ חישובי ערך, ותעביר לשיטות הנקודות. שטחוב את הכפל ב($O(n)$ זמן, אח"כ תבצע אינטרופולציה חזקה שתעלה ב($O(n^2)$ וסימית. סה"כ עליה לך $O(n^2)$. Cut נראה שיטה, שתאפשר את המעבר הראשון והאחרון ב($O(n \log n)$ זמן, מה שיחפה את האלגוריתם לזמן $O(n \log n)$. כיצד? נרצה לבחור ערכי x -ים ספציפיים מאוד.

המטרה: נרצה לחשב את $(x) A$ בה ערכים שונים - x_0, \dots, x_n . מדוע לא $2 - n$? קל להסביר n , ואח"כ קל להכליל את הרעיון וברור ששאיפוטות זו אותה סיבוכיות.

הנחה: n הוא חזקה של 2. ניתן להתגבר על הנחה זו, באופן שלא יגע בסיבוכיות, אך בשביל הפשטות המתמטית נניח הנחה זו. מדוע ניתן להניח זאת? אפשר להושך מקדים של אפס ואז תמיד נקבל חזקה של 2.

נגדיר את הפולינומים הבאים:

$$A_{even}(x) = a_0 + a_2x + a_4x^2 + \dots + a_{n-2}^{\frac{n}{2}-1}$$

$$A_{odd}(x) = a_1 + a_3x + a_5x^2 + \dots + a_{n-1}^{\frac{n}{2}-1}$$

טעינה:

$$A_{even}(x^2) + xA_{odd}(x^2) = A(x)$$

$$A_{even}(x^2) - xA_{odd}(x^2) = A(-x)$$

נשים לב כי $A(x)$ הם מדרגות חסומה $\frac{n}{2}$, שזה חצי הגדרה החסומה של $A(x)$.

נסיוון ראשון: נחשב את A_{odd} ב- n ערכי x שונים: x_0^2, \dots, x_n^2 . נחשב את A_{even} ב- n ערכי x שונים: x_0^2, \dots, x_n^2 . ואז נשתמש בנוסחה לעיל כאן בטענה, $A_{even}(x^2) + xA_{odd}(x^2) = A(x)$. נשים לב כי החישוב בסוף עולה $O(n)$ שהרי מחשבים n ערכים, ובכל קראייה אנחנו קוראים $\lceil \frac{n}{2} \rceil$ פעמיות" ולכוארה רקורסיבית אפשר לקבל את הנוסחה הבאה $n + T(n) = 2T(\frac{n}{2}) + O(n)$ ולפי משפט האב, $T(n) = O(n \log n)$

זה לא עובד. למה?

1. הובטה כי $x_{n-1}, \dots, x_0^2, \dots, x_{n-1}^2$ שונים. אך מי אמר שה- $x_3 = -10$, $x_3 = 10$, $x_2 = x_2^2 = x_3^2 = 100$ וアイם שונים.
2. בשימוש בהפרד ומשול, מובטה לכך כי סוג תת בעיה שתתקבל יהיה זהה לבעיה המקורית רק על קלט קטן יותר. בעיה המקורית, נדרש לחשב n ערכים של A , באופן שיעלה $O(n)$. נשים לב כי הפולינום חסום מדרגה n בהתחלה, ואז מחשבים בהתאם n ערכים. אוח"ב לאחר שמסתכמים על $\frac{n}{2}$, הפולינום חסום מדרגה $\frac{n}{2}$ אבל גם כאן אנו נדרשים לחשב n ערכים שונים. וכן הלאה – זו לא תת בעיה.

2.3.1 תוכנות הנגידיות החלשה

יהי k חזקה של 2. לסדרת ערכי האיקס: $(x_0, x_1, \dots, x_{k-1})$ יש את תוכנות הנגידיות החלשה אם אחד מהתנאים הבאים מתקיים:

- א. $k = 1$.
- ב. $\forall 0 \leq j \leq \frac{k}{2} - 1 : x_{\frac{k}{2}+j} = -x_j$

דוגמה. הסדרה $-2, -5, 1, -3, -1, 3, 2, -5$ היא בעלת תוכנות הנגידיות החלשה, נשים לב שהחצי השני הוא הנגדי של החצי הראשון.

נשים לב – כאשר נעה את איברי הסדרה בריבוע, נקבל כי החצי הראשון של הסדרה שווה לחצי השני.

נסיוון שני: נניח כי מטרת העל החדשת שלנו, היא לחשב את A ב- n ערכי x שימושיים את תוכנות הנגידיות החלשה. מכאן ש: $(x_j)^2 = (-x_j)^2$. לכן, הסדרה החדשת שלנו היא מורכבת משתי סדרות זהות שבאות אחת אחורי השניה. לכן אם נחשב את A על החצי הראשון, אין צורך לחשב על החצי השני כי קיבלו אותו בחינם". האלגוריתם החדש:

- א. נחשב את A_{odd} ב- $\frac{n}{2}$ ערכי x : $x_0^2, x_1^2, \dots, x_{\frac{n}{2}-1}^2$
- ב. נחשב את A_{even} ב- $\frac{n}{2}$ ערכי x : $x_0^2, x_1^2, \dots, x_{\frac{n}{2}-1}^2$
- ג. לכל $1 \leq i \leq \frac{n}{2}$ יתקיים: $A(x_i) = A_{even}(x_i^2) + x_i A_{odd}(x_i^2)$
- ד. לכל $1 \leq i \leq \frac{n}{2}$ הACHI השני של הערכים נשים לב כי לפי הנגידיות החלשה ומעבר שראיינו לעיל מתקיים: $A(x_{\frac{n}{2}+i}) = A(-x_i) = A_{even}(x_i^2) - x_i A_{odd}(x_i^2)$, וסה"כ באמצעות א'ב חישבנו גם את החצי הוה של הערכים בלי לבצע פעולות נוספת.

סה"כ חישבנו את n הערכים הפעם, לכוארה ללא בעיה שם יփכו לשוניים, לכוארה באופן רקורסיבי ניתן שוב לטעון $(n \log n) + n = O(n \log n)$. זה שוב לא עובד – זה שוב לא עובד – זה שוב לא תת בעיה! המטרה שלנו הייתה לחשב סדרות ערכים שימושיים את תוכנות הנגידיות החלשה. לאחר העלאה בריבוע, הם לא מקיימים את תוכנות הנגידיות החלשה. ואי אפשר לומר שזו תת בעיה.

2.3.2 תוכנות הנגידיות החזקה

יהי חזקה של 2. לסדרת ערכי האיקס: $(x_0, x_1, \dots, x_{k-1})$ יש את תוכנות הנגידיות החזקה אם אחד מהתנאים הבאים מתקיים:
 א. $k = 1$.
 ב. לסדרה יש את תוכנות הנגידיות החלשה, וגם לסדרה $(x_0^2, x_1^2, \dots, x_{\frac{n}{2}-1}^2)$ יש את תוכנות הנגידיות החזקה (באופן רקורסיבי).

בעת נשים לב - כי הטעיה הייתה לנו מוקודם נפתרה למורי. להלן האלגוריתם:
המטרה: לחשב את A מדרגה חסומה n בא ערכי x שונים שקיימים את תוכנות הנגידיות החזקה.

- א. נחשב את A_{odd} שהוא מדרגה חסומה $\frac{n}{2}$, ב $\frac{n}{2}$ ערכי x : $(x_0^2, x_1^2, \dots, x_{\frac{n}{2}-1}^2)$ מההגדרה הרקורסיבית, ערכים אלו מקיימים את תוכנות הנגידיות החזקה.
- ב. נחשב את A_{even} שהוא מדרגה חסומה $\frac{n}{2}$, ב $\frac{n}{2}$ ערכי x : $(x_0^2, x_1^2, \dots, x_{\frac{n}{2}-1}^2)$ מההגדרה הרקורסיבית, ערכים אלו מקיימים את תוכנות הנגידיות החזקה.
- ג. לכל $1 \leq i \leq \frac{n}{2}$ יתקיים: $A(x_i) = A_{even}(x_i^2) + x_i A_{odd}(x_i^2)$
- ד. לכל $1 \leq i \leq \frac{n}{2}$ הACHI השני של הערכים נשים לב כי לפי הנגידיות החלשה ומעבר שריאנו לעיל מותקים: $A(-x_i) = A_{even}(x_i^2) - x A_{odd}(x_i^2)$, וסה"כ באמצעות א+ב חישבנו גם את הACHI הזה של הערכים בלי לבצע פעולות נוספת.

זה"כ הטעיות נפתרו - בכל שלב אנחנו מקבלים תות בעיה, וכן הערכים תמיד יקימו את תוכנות הנגידיות החזקה. זה"כ סיבוכיות הזמן של האלגוריתם הינה $O(n) = 2T(\frac{n}{2}) + O(n)$, וממאנstre נקבל $O(n \log n)$ למעבר מממדים ליצוג ע"י נקודות.

2.3.3 איזה מספרים מקיימים את תוכנות הנגידיות החזקה?

מספרים מרוכבים. נזכר כי מס' מרוכב ניתן לייצג ע"י $a + bi = rcis\theta = re^{i\theta}$. אנחנו נתמקד במספרים בהם $r = 1$.
 מספר ω נקרא שורש היחידה המרוכב מסדר n , אם $1 = e^{\frac{2\pi i}{n}}$. למשל, נראה כי $\omega^n = 1$, מותקים כי $\omega^8 = 1$.

נדיר את המספר ω להיות: $e^{\frac{2\pi i}{n}} = \omega_n$. נשים לב כי תמיד $\omega_n^n = 1$. (הURA - נשים לב כי $i\omega_4 = e^{\frac{\pi i}{2}} = i$. מדוע? נראה כי האוזית היא $\frac{\pi}{2}$, כלומר 90 מעלות. אם נזוז 90 מעלות מהכיון החובי של הציר המשמי, נגיע בדיק למספר i . וכך בדיק מחשבים מספרים אלו).

n שורשי היחידה מסדר n הינם: $\omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}$.

טענה: לכל $0 \leq k \leq n$ מותקים כי $(\omega_n)^k$ הוא שורש ייחידה מסדר n .
הוכחה: נראה להוכיח כי מספר זה בחזקת n שווה לאחד. ובכן -

$$((\omega_n)^k)^n = ((e^{\frac{2\pi i}{n}})^k)^n = (e^{\frac{2\pi i n k}{n}}) = e^{2\pi i k} = e^0 = 1$$

טענה 2: יהי $n > 1$ חזקה של 2. אז לכל $0 \leq k \leq \frac{n}{2} - 1$ מותקים $\omega_n^{\frac{n}{2}+k} = -\omega_n^k$
הוכחה:

$$\omega_n^{\frac{n}{2}+k} = \omega_n^{\frac{n}{2}} \times \omega^k = \omega^k \times e^{\frac{2\pi i \frac{n}{2}}{n}} = -\omega^k$$

טענה 3: יהיו $1 < n$, חזקה של 2. איזי הריבועים של $\omega_n^0, \omega_n^1, \dots, \omega_n^{n-1}$ הם בדיקת $\frac{n}{2}$ שורשי היחידה מסדר $\frac{n}{2}$. (כלומר, נעלם אותם בריובע, נקבל בדיקת שורשי היחידה מסדר $\frac{n}{2}$, וכל אחד מהם יופיע פעמיים - כלומר יהיה כפליות).

הוכחה: עבור $1 \leq k \leq \frac{n}{2} - 1$ מתקיים

$$(\omega_n^k)^2 = e^{i\frac{2\pi}{n}2k} = (e^{\frac{i2\pi}{2}})^k = \omega_{\frac{n}{2}}^k$$

טענה 4: יהיו $1 \geq n$ חזקה של 2. סדרת n שורשי היחידה מסדר n : $\omega_n^0, \omega_n^1, \omega_n^2, \dots, \omega_n^{n-1}$ מקיימים את תכונת הנגידיות החזקה.

נשים לב - n שורשי היחידה מסדר n הם אכן שונים זה מזה.

2.4 האלגוריתם FFT

המטרה: לחתות את המערך עם המקדים של A , ולהשאבת הפולינום A ב- n שורשי היחידה מסדר n (אשר הוכחנו שמקיימים את תכונת הנגידיות החזקה). כלומר לחשב את $(A(\omega_n^0), \dots, A(\omega_n^{n-1}))$.

קלט: $A = (a_0, a_1, \dots, a_{n-1})$ מדרגה חסומה n

א. אם $n = 1$, החזר a_0 .

ב. כתעת נגידיר את $A_{even} = (a_0, a_2, \dots, a_{n-2})$ מדרגה חסומה $\frac{n}{2}$.

ג. כתעת נגידיר את $A_{odd} = (a_1, a_3, \dots, a_{n-1})$ מדרגה חסומה $\frac{n}{2}$.

ד. כתעת נתחיל את הרכושה: $P_{even} = FFT(A_{even})$, כאשר P_{even} יפעיל את FFT על A_{even} - כלומר מחשבים את A_{even} ב- n שורשי היחידה מסדר $\frac{n}{2}$.
כלומר, מה שחוור מהרכושה הינו $[A_{even}(\omega_{\frac{n}{2}}^0), A_{even}(\omega_{\frac{n}{2}}^1), \dots, A_{even}(\omega_{\frac{n}{2}}^{\frac{n}{2}-1})]$

ה. בדומה - $P_{odd} = FFT(A_{odd})$.

ו. החל מ-0 עד $j = \frac{n}{2} - 1$ בצע: (כתעת אנחנו רוצים לחשב את הפלט שלנו - מחזירים לבסוף וקטוּר \vec{y} עם חישוב הערכים בהתאם לפי הנוסחאות שראינו)

$y_j = P_{even}[j] + w_n^j \times P_{odd}[j]$.
נשים לב כי $y_j = A(\omega_n^j)$ ולפי נוסחה שראינו $A_{even}(x^2) + xA_{odd}(x^2) = A(x)$, כמו כן ניתן להמירה בהתאם $A_{even}(\omega_n^{j/2}) + \omega_n^j A(\omega_n^{j/2}) = A(\omega_n^j)$ לפי טענה 3 לעיל, ולכן $A_{even}(\omega_{\frac{n}{2}}^j) + \omega_n^j A(\omega_{\frac{n}{2}}^j) = A(\omega_n^j)$
זה ממש שקול ל- $(P_{even}[j] + w_n^j \times P_{odd}[j])$.

$$y_{\frac{n}{2}+j} = P_{even}[j] - w_n^j \times P_{odd}[j].2$$

ג. כהארקורסיה נגמרה - החזר את (y_0, \dots, y_{n-1})

סיכום זמן הריצה: הנוסחה $- T(n) = 2T(\frac{n}{2}) + O(n) = O(n\log n)$, ולכן סה"כ זמן חישוב האלגוריתם שמקבל פולינום בשיטת המקדים, וממיר אותו n נקודות ספציפיות (שורשי היחידה מסדר n) לפי שיטת הנקודות הוא $O(n\log n)$.

2.5 כיצד נעבור בעת מושית הנקודות חוזרת לשיטת המקדים?

נשים לב כי אנחנו יודעים את ערכי x הנקודות שלנו, הם n שורשי היחידה מסדר n .
 $x_0 = (\omega_n)^0 = 1, x_1 = (\omega_n^1), \dots, x_{n-1} = (\omega_n^{n-1})$.

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n^1 & (\omega_n^1)^2 & \dots & (\omega_n^1)^{n-1} \\ 1 & (\omega_n^2)^1 & (\omega_n^2)^2 & \dots & (\omega_n^2)^{n-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 1 & (\omega_n^{n-1})^1 & (\omega_n^{n-1})^2 & \dots & (\omega_n^{n-1})^{n-1} \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}$$

נשים לב כי בידינו קודם לכך היה \vec{a} , כפלנו אותו במטריצת FFT שלו, V וקיבלו את \vec{y} . באופן כללי - כפל נאייבי של מטריצה בוקטור עולה $O(n^2)$ זמן.

מסקנה חשובה (!!): כפל של מטריצת ונדרמונייה שמודגרת ע"י n מספרים שמקיימים את תוכנת הנגידות החזקה, בוקטור \vec{a} עולה $O(n\log n)$ (שהוא בדיקו תהליך שעשה האלגוריתם).

טענה: המטריצה ההופכית של V הינה המטריצה:

$$V^{-1} = FFT^{-1} = \frac{1}{n} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n^{-1} & (\omega_n^{-1})^2 & \dots & (\omega_n^{-1})^{n-1} \\ 1 & (\omega_n^{-2})^1 & (\omega_n^{-2})^2 & \dots & (\omega_n^{-2})^{n-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 1 & (\omega_n^{-(n-1)})^1 & (\omega_n^{-(n-1)})^2 & \dots & (\omega_n^{-(n-1)})^{n-1} \end{pmatrix}$$

אם נסתכל על המטריצה $\vec{a} \times FFT^{-1}$ (שהרי נרצה להכפיל בא' כי נשים לב $\frac{1}{n}$ שיצא החוצה מהמטריצה), נראה כי היא מטריצת ונדרמונייה על הערכים: $(\omega_n^0, \omega_n^{-1}, \omega_n^{-2}, \dots, \omega_n^{n-1})$.

מסקנה: על מנת לעבור מוקטור \vec{y} למקטור $\vec{a} = FFT^{-1} \times \vec{y}$ מתקבלים מקבילים ממש $\vec{y} = FFT^{-1} \times \vec{a}$, או מכפלה של מטריצת ונדרמונייה על n מספרים שמקיימים את תוכנת הנגידות החזקה (גם, n שורשי היחידה מסדר n), בוקטור, ראיינו בטענה לעיל שמכפלה זו עולה $O(n\log n)$, וכך המסקנה **שלאו היא שוגם המעבר חוזרת אל שיטת המקדים, עולה גם הוא $O(n\log n)$** .

סיכום - כפל פולינומיים:

- א. מקבלים את הפולינומיים $A(x), B(x)$ המוצגים ע"י מקדים.
- ב. בעזרת אלגוריתם FFT , בזמן $O(n\log n)$ מקבלים את $A(x), B(x)$ מוצגים ע"י n נקודות (שהם שורשי היחידה מסדר n)

- ג. מכפילים את שני הפולינומים בזמן $O(n)$ בשיטת הנקודות, כיון שהם מיוצגים ע"י אוטם ערכי x (שורשי היחידה).
- ד. משתמשים שוב ב- FFT , באמצעות הכפלת הערך ע"י המטריצה FFT^{-1} שגם היא מטריצה נדרומונדה, מחזירים את הפולינומים לשיטת המקדים, מה שיעלה עוד $O(n \log n)$. סה"כ - $O(n \log n)$ לכפל פולינומים.

2.6 תרגילים FFT

2.6.1 כפל פולינומים מדרגות חסומות שונות

$$\text{קלט: } m < A(x) = \sum_{i=0}^{n-1} a_i x^i, B(x) = \sum_{i=0}^{n-1} b_i x^i \text{ באשר בה"כ } n \\ \text{פלט: } C(x) = A(x) \times B(x)$$

פתרונות:

1. **אפשרות ראשונה:** נתיחס אל B כפולינום חסום מדרגה n . לפי FFT - כפל שני פולינומים מדרגה זו יעלה $(O(n \log n))$.
2. **אפשרות שנייה:** נוכל להשתמש באלגוריתם נאיבי לכפל פולינומים - שעובר עליהם אחד אחד לפי הנוסחה, ולקבל סיבוכיות $O(nm)$.
3. **אפשרות שלישיית:** להרץ במקביל את אפשרות 1 ואפשרות 2, ונקבל סיבוכיות $\min(m, \log n))$.
4. **אפשרות רביעית:** נרצה להגיע לזמן ריצה $O(n \log m)$ הרעיון יהיה לחלק את A לפולינומים קטנים יותר בגודל m , סה"כ $\frac{n}{m}$ פולינומים. נשים לב כי:

$$A = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \dots + a_{n-1} x^{n-1} =$$

$$(a_0 + a_1 x + \dots + a_{m-1} x^{m-1}) + (a_m x^m + \dots + a_{2m-1} x^{2m-1}) + \dots + (a_{n-m} x^{n-m} + \dots + a_{n-1} x^{n-1})$$

$$(a_0 + a_1 x + \dots + a_{m-1} x^{m-1}) + x^m (a_m + \dots + a_{2m-1} x^{m-1}) + \dots + x^{n-m} (a_{n-m} + \dots + a_{n-1} x^{m-1})$$

icut נשים לב, כל בלוק מהן"ל יסומן $A_{km} = a_{km} + \dots + a_{km+m-1} x^{m-1}$ ונקבל:
 $\sum_{i=0}^{\frac{n}{m}-1} A_i(x) \cdot x^{mi}$
 icut נראה כי:

$C(x) = A(x) \cdot B(x) = A_0(x)B(x) + x^m B(x)A_1(x) + \dots + x^{n-m} B(x)A_{\frac{n}{m}-1}(x)$
 icut, המכפל של פולינום $C(x)$ מורכב מ- $\frac{n}{m}$ כפליים שונים של פולינומים, כל אחת מהפולינומים הינו בדרגה m (פולינום A_{ki} מדרגה m וכן פולינום B מדרגה (m))
 סה"כ סיבוכיות זמן הריצה תהיה -

$$\frac{n}{m} \times m \log m = O(n \log m)$$

2.6.2 בעיית 3SUM

פלט: מערך $A = [a_0, \dots, a_{n-1}]$ באשר

$$a_k = a_i + a_j, a_k \text{ כך ש } a_i, a_j \in A$$

פתרונות ראשוני:

האלגוריתם יפעל כך: הרגע ייה ל הגיע לסיבוכיות של $O(n^2)$. האלגוריתם יפעל כך -

א. נמיין את המערך A

ב. עבור $0 \leq k \leq n-1$ בצע:

1. הגדר $a_k = a_i + a_j$

2. אם $a_k = a_i + a_j > a_k$ אז שוכם המספרים קטן מדי, לכן צריך להוריד מס' מינימום, ככלומר:

$j = j - 1$

3. אם $a_i + a_j < a_k$ אז שוכם המספרים גדול מדי, לכן צריך לתקדם משמאל לימין. ככלומר:

$i = i + 1$

ג. אחרת - החזר *false*

$$\text{זמן הריצה: } T(n) = \sum_{k=0}^{n-1} n + O(n \log n) = O(n^2)$$

פתרונות שני (מספרים מעולים חסומים):

כעת נניח כי כל המספרים $a_i \in A$ הינם חסומים בתחום $[1, 10n^{1.5}]$

נרצה לקודד את המערך A לפולינום מדרגה חסומה $10n^{1.5}$ בזורה הבא:

המקדם של x^i בפולינום $C(x) = a_i$ אם x^i יהיה שווה 1,吟 $i \in A$,吟 0 אחרת. למשל, עבור $A = [2, 5, 7]$ נקבל

$$A(x) = x^2 + x^5 + x^7$$

כעת: האלגוריתם יבצע את הכפל של הפולינום בעצמו, כלומר $A^2(x)$. לאחר מכן, נחפש מקומות מכפלה של שני פולינומים $C(x) \cdot C(x) = A^2(x)$, בו המקדם $2, c_i \geq 1$,吟 $i \leq n$ והמקדים כי המקדם a_i במערך A שווה 1. אם האלגוריתם מצא מקום שכזה, הוא יחזיר 1. אחרת יחזיר שקר.

נכונות: מוכיח חזקיות $x^i x^j = x^{i+j}$, כאשר נכפול את הפולינום שמייצג את המערך בעצמו, פולינום המכפלה יציג למשהו סכומים של איברי המערך. מכיוון, כל חזקה בפולינום חזקתו ≤ 2 הוא בהכרח מכפלה של שני מספרים אחרים במערך. כעת, כל שנותר לעשות לאלגוריתם הוא לעבור במערך המקורי, ולחפש מס' i שקיים בו. אם מצאנו צזה - קיימים מספר שווה לסכום שניים אחרים.

סיבוכיות זמן הריצה: כפל לפי FFT יעלה $(n^{1.5} \log n)$, מעבר על פולינום המכפלה ובדיקה מהמקדמים שלו יעלה $(O(n^{1.5} \log n))$, וסה"כ סיבוכיות זמן הריצה תהיה $O(n^{1.5} \log n)$.

2.6.3 בעית חישוב מרחק האמינו

בעית התאמת המחרוזות מוגדרת כך:

קלט: מחרוזת T באורך n - "טקסט", ומחרוזת P באורך $m \leq n$ - "תבנית".

פלט: כל המיקומות ב' T ' שモופיע בהם.

מרחק האמינו:

לשתי מחרוזות A, B באורך זהה n , מרחק האמינו של A ו- B הוא מס' האינדקסים בהם A ו- B שונים. ככלומר

$$HD(A, B) = |\{i | A[i] \neq B[i]\}|$$

נדיר כעת את הבעיה הבאה שמקלילה את בעית התאמת המחרוזות:

בעית חישוב מרחקי האמינו בין טקסט לתבנית:

קלט: מחרוזת $[n]$ באורך n : "טקסט". ומחרוזת $[m]$ באורך m : "תבנית".
 פלט: לכל היסט $i \leq n - m$ את מරחק ההאמיניג $[m]$ $HD(P, T[i+1, \dots, i+m])$ (נשים לב - הפלט יוחזר במערך בגודל $m - n$, כאשר בתא הראשון יופיע מרחיק ההאמיניג עבור המחרוזת שהתחילה בתא הראשון והסתירה על התאים $m - 1, \dots, m - 2$ וכן הלאה. התא האחרון יהיה במיקום $m - n$ שהוא יבדוק את המחרוזות שהתחילה במיקום $m - n$ והסתירה בהמשך במיקום n . נחשב בעיה זו כאשר הא"ב הוא בינארי, כלומר מעל $\{0, 1\}$.

פתרון נאיבי - נבצע בדיקה של כל היסט באופן נאיבי ע"י בדיקה שתעללה $O(m)$, יש $C(n)$ היסטים וכך הפתרון יעלה $O(nm)$.

פתרונות באמצעות FFT:

נסמן את הטקסט באוויות $T = a_0a_1\dots a_{n-1}$ וכן $P = b_0b_1\dots b_{m-1}$ מאפשר לכפול פולינומים. ננסה לחושב על הכפל בצורה מעטה יותר מופשטת - כפולה ביןירות הפעולות בין שני אובייקטים ומזהירה מספה. נראה כי כפל פולינומים, דומה לכפל בין מספרים מרובי ספרות - המתאים לשפה זו (כלומר a_i , המקדם של הבסיס בחזקת i). ההבדל המשמעותי היחיד - הוא שבכפל בין ספרות יש בסיס, וכאשר המקדם של ספרה גדול מהבסיס הוא עובר לנשא לחזקהhabah של הבסיס. כפל בין מספרים ניתן לחשב באמצעות כפל ארוך, ונשים לב שאותו יציג יבוד גם לכפל פולינומים. כמו כן, FFT אינו כפל ארוך, אך סוף סוף תוצאות הכפל זהה לא משנה האם נעשו באמצעיות כפל ארוך או באופן מהיר יותר באמצעות אלגוריתם FFT. נסתכל על כפל ארוך בין שני מספרים/פולינומים המייצגים את הטקסט והתבנית. נכפול את T ב- P^R (היפוך סדר המקדמים).

דוגמה: אם $n = 5$ ו- $m = 3$ נסתכל על הפולינומים (b_2, b_1, b_0) ו- $(a_0, a_1, a_2, a_3, a_4)$ וכפלו אותם בגישה כפלי ארוך כדלקמן -

$$\begin{array}{ccccc} a_0 & a_1 & a_2 & a_3 & a_4 \\ & & b_2 & b_1 & b_0 \\ \hline & a_0b_0 & a_1b_0 & a_2b_0 & a_3b_0 & a_4b_0 \\ a_0b_1 & a_1b_1 & a_2b_1 & a_3b_1 & a_4b_1 \\ a_0b_2 & a_1b_2 & a_2b_2 & a_3b_2 & a_4b_2 \end{array}$$

ניתן לשים לב שבסכום אחד משולשת העמודות האמצעיות (שבכווות) נקבל סכום של אוויות מקבילות בתבנית ובטקסט, כאשר שמיים את התבנית מול הטקסט בהיסטים שונים. ככלומר - קיבלנו את כל אפשרות הטקסט עבורם בקלט הנוכחי. נרצה שהסכום הנ"ל יהיה משמעותית. ככלומר: נרצה שבסכום אחד מרחיק ההאמיניג, כל שנצטרך יהיה לחבר את כל הערכים במסלול האדום ולקיים את מרחיק ההאמיניג, בוורוד וכן בכחול ופתרנו את הבעיה. נדאג לכך שהסכום יהיה בדיקת מס' המוקומות בהם התבנית מתאימה לטקסט.

כלומר, נרצה שכפל בין שני תווים ייגז $ab = 1$ אם $a = b$ וכן $ab = 0$ אחרת. נראה כי פעולה זו ניתנת לייצוג ע"י הtablלה -

$T \setminus P$	0	1
0	1	0
1	0	1

פעולה זו עולה $O(1)$ לחישוב. עם זאת, נראה כי חישוב המכפל הארוך יעלה $O(nm)$. לא יותר טוב מהדרך הנאיתית.

נרצה להשתמש ב- FFT בשיביל לבצע את האלגוריתם בזמן $O(n \log m)$. עם זאת, FFT מוגדר ככפל פולינומיים מעל \mathbb{C} , ככפל רגיל, לא כפעולה שתיארנו לעיל. مكان שנצטרך להתגבר על בעיה זו: נפריד את החישוב לשני חישובים נפרדים.

רצינו לספר את כל המיקומות בהם התו בתבנית זהה לתו בטקסט בהיסט המותאים. כל מקום כאז עונה על בדיק אפשרות אחת מתוך שתיים:

$$\begin{array}{l} a = b = 1 \\ a = b = 0 \end{array}$$

נספר כל אחד מהמקרים הללו בנפרד, ונסכום.

לספרת המיקומות בהם $a = b = 1$ נשמש בפעולת הבהא:

$T \setminus P$	0	1
0	0	0
1	0	1

זהי בדיק פועלות המכפל הרגילה. לכן נוכל להשתמש ב- FFT באופן מיידי

לספרת המיקומות בהם $a = b = 0$ נשמש בפעולת הבהא:

$T \setminus P$	0	1
0	1	0
1	0	0

כדי לקבל פעולה זו באמצעות מכפל רגיל, נהפוך את הביטים בטקסט ואת הביטים בתבנית - כל בית שהוא 0 יהיה 1 וכל מי שהוא 1 יהיה אפס. וכך נקבל עבור $a = b = 0$

$\bar{T} \setminus \bar{P}$	1	0
1	1	0
0	0	0

זו בדיק פועלות המכפל.

לסיכום: באמצעות שתי פעולות FFT על הקלטים של הפולינומים ופעולות המכפל שהוגדרו לעיל, נדע כמה התאמות יש בין התבנית לטקסט בכל היסט. מה שיוחזר לנו בתוצאה מהפעלת FFT לבדיקה התאמות על 1 יוצג במערך מקדמים A_1 , ומה שיוחזר בתוצאה מהפעלת FFT לבדיקה התאמות על 2 יוצג במערך מקדמים A_2 , וסה"כ נבצע חיבור למערך חדש $C[i] = A_1[i] + A_2[i]$ לקבלת מס' ההתאמות. לבסוף, נכח את מספר ההתאמות ונחסר אותו מ- m , ונCallCheck את מספר האי התאמות שהוא בדיק מרחק האמיגג. ככלומר, לכל תא i נגדיר $C[i] = m - C[i]$.

כפי שראינו בדוגמה 1: פולינומים מדרגות מסוימות, ניתן לכפול שני פולינומים שיחסומים בדרגות $m \geq n$ בעלות $O(n \log m)$ וכן או סיבוכיות האלגוריתם. חיבור הפולינומים יעלת $O(n)$ ולא ישנה את הסיבוכיות האסימפטוטית. וכן, ביצוע פעולה NOT בשיביל שנוכל להשתמש ב- FFT על התאמות של 0 עליה $m + n$. סה"כ סיבוכיות אלגוריתם - $O(n \log m)$.

נשים לב: ניתן להרחיב את הפתרון גם לא"ב שהוא לא רק $\{0, 1\}$. ניתן להרחיב את הפתרון לארסה טובה יותר מאשר NOT - נניח ונקבל א"ב $\{0, 1, 2\}$. תחילת - נכתב 1 בכל המיקומים של 2 ובאשר המיקומים כתוב אפס ונבצע FFT ונדע מהו מס' ההתאמות של 2.

לאחר מכן נכתב 1 בכל המיקומים של 1 ובשאר כתוב אפס וכן הלאה כנ"ל על אפס. מכאן קיבלנו מסקנה: בהינתן א"ב Σ , נוכל לבצע את אלגוריתם מרחק ההאמינג עלולות כולל של $O(n + m + n \log m)$ שכן נדרשים לבצע בכל שלב $m + n$ על מנת להפוך את המערך לאחדות ואפסים, וכן עוד $n \log m$ לביצוע ה- FFT .

להלן האלגוריתם:

א. צור מערך חדש M בגודל $n - m + 1$

ב. לכל $\sigma \in \Sigma$:

ספר לכל היסט אפסי של התבנית בטקסט את מספר ההתאמות של התו σ ע"י FFT של $T_\sigma \times P_\sigma^R$.
ג. החזר את M .

נשים לב כי ככל היוטר במצב בו כל התווים שנקל שווים, יתקיים
 $|\Sigma| = O(m + n) = O(n)$

3 הרצתה 2 - MST

3.1 עץ פורש מינימום

עץ פורש מינימום, או *MST* (Minimum spanning tree) הוא מה שנעסוק בו בהרצאה זו. למה צריך גראפים? למשל, עבור רשתות תקשורת. כל קודקוד בעץ הוא מוחשב בראשת, וכל קשר בין הקודקודים מציג את תקשורת ישירה בין שני המוחשבים בראשת. אנחנו חברה, שרצה למכוון תקשורת, והמטרה שלנו היא שהקשרות שנבחר ישאירו את הגרף קשיר. ככלומר - נרצה לבחור קשותות כרצוננו רק נשים לב שבכל שלב נתנו נוכל להגיע לכל מוחשב בראשת. נשים לב - שבמהלך הבחירה שלנו יתכן ונבחר קשותות נוספות, בשליליה לא להגיע למצב שהגרף לא קשור.

עץ פורש: תת גראף של הגרף המקורי, שהוא קשור ולא מעגלים (עץ). ככלומר $V' = V \setminus E'$.

יצוג לגרפים: מטורצת שכניות, רשימה (*List*) של שכניות.

נשים לב כי נרצה למצוא את העץ הפורש הנ"ל שיאפשר לנו למצוא דרך להגעה לכל המוחשבים (קודקודים) בראשת (graft). נשים לב - כי יתכן ויהיה כמה דרכים לעשות זאת, כמו כן - יתכן ולכל מעבר בראשת מסוימת יהיה מחיר שונה (כלומר, לכל קשר יהיה מחיר ששולם שנעלה עליו). וכך כל קשר תסומן אצלנו בערך מספרי מסוים.

בהינתן x קודקודים בעץ הפורש, נרצה למצוא $1 - x$ קשותות שהעלות שלן יחד היא הנמוכה ביותר.

נשים לב כי באשר נעבד עם עצים פורשים מינימום, נניח מראש כי הגרף יהיה קשור.

יהי $G = (V, E)$ מולטי גראף קשור ממושקל עם פונקציית משקל על הקשותות $w : E \rightarrow \mathbb{R}$.
יהי $T = (V, E_T)$ עץ פורש של G . אז, נאמר שהמשקל של T הוא

$$w(T) = \sum_{e \in E_T} w(e)$$

עץ פורש מינימום של G הינו עץ פורש שמשקליו הוא הקטן ביותר.

מסקנה: יהי T עפ"ם, אז לכל עץ פורש $T' \in G$ מתקיים $w(T') \geq w(T)$.

הערה: מולטי גראף הוא גראף בו קבוצת הקשתות הינה *multi-set*, כלומר אין שני קוזקווים בגרף, יכולות לעבור מספר קשトラות, ונדוע ש格將ה להשתמש בכך הרוי בגרף כי כאשר תחפש עפ"ם, בהיותו שני קוזקווים u, v וקשתות e_1, e_2 , שמשקלן בהתאם 1, 2, נרעה לבחורו בקש שמשקללה 1. בהמשך, נזכיר מזוז האלגוריתם פועל על מולטי גראף למטרות זהה לא נראה אינטואטיבי.

הערה: גראף לא פשוט הוא גראף בו התנועה דו כיוונית, אס קיימת $u \rightarrow v$: כלומר לא ניתן לעבור מ v ל u . גראף פשוט הוא זה בו התנועה מוגדרת בכיוון מסוים. ככלומר זה שקיימת $u \rightarrow v$ לא גורר שייתנו ללכט בכיוון $v \rightarrow u$.

3.2 בעיית מציאת עץ פורש מינימום

קלט: גראף לא פשוט קשיר $G = (V, E)$ ופונקציית משקל $w : E \rightarrow \mathbb{R}$

פלט: עץ פורש מינימום של G (ביחס לפונקציית משקל w).

3.3 אלגוריתם חמדני (Greedy)

מקבלים החלטה לקלאלית וממשיכים ורקורטיבית בלי לשנות את ההחלטה ובלי לדעת מה הפתרון ברקורסיה. למשל - חיפוש ביןארי.

למה הבחירה החמדנית: הבחירה שהאלגוריתם ביצ' באופן חמדני לא מונעת ממנו להגיע לפתרון האופטימלי.

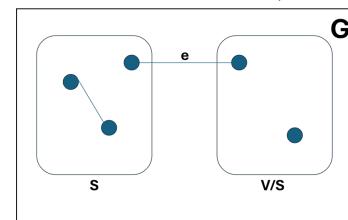
למה תת המבנה האופטימלי: מכלול בחירות חמדניות יביא את התוצאה האופטימלית.

3.4 למת הבחירה החמדנית

חתך: יהי $G = (V, E)$ גראף וכי $S \subset V$ כך ש $S \neq \emptyset$. החלוקה $(S, V/S)$ נקראת חתך של G וקשתות $\{e = (u, v) | u \in S, v \in V/S\}$ נקראות **קשתות שחוצה את החתך / קשתות בחתך**.

נשים לב - S עשוי לא להיות שווה V , ולעומת לא תהיה ריקה.

בתמונה לעיל, e היא קשת שחוצה את החתך.



קשת קלה ביותר בחתך: יהי $G = (V, E)$ גראף לא פשוט עם פונקציית משקל w . יהי $(S, V/S)$ חתך של G . קשת $e \in E$ נקראת קשת קלה ביותר בחתך $(S, V/S)$ אם לכל קשת e' שחוצה את החתך מתקיים $w(e') \leq w(e)$. נשים לב, ניתן שיש כבוח קשתות כללו באותו משקל שכן הקלות ביותר.

лемה 1: יהיו $G = (V, E)$ מולטי גרף קשור עם פונקציית משקל $w : E \rightarrow \mathbb{R}$. לכל $S \subset V \neq \emptyset$, לכל קשת e קלה בחתך $(S, V/S)$ קיים עפ"מ שמקל את e .

הוכחה: יהיו $G = (V, E)$ מולטי גרף קשור עם פונקציית משקל $w : E \rightarrow \mathbb{R}$.

יהי $(S, V/S)$ חתך של G , ותהי $e = (u, v) \in S, v \in V/S$. קשת קלה בחתך, כאשר $u \in S, v \in V/S$. (e כהכרח קיים נזה, כיון ש- G קשור, ולכן יש לו עזים פורשים, ובהכרח אחד מרטס מינימלי). אם $e \in T$ אז סימנו.

אחרת, $T \notin e$. לפי תכונות של עזים - קיוס מסלול פשוט P (כל הקוזקודות בו, ופעמים אחת בלבד) יחוֹז C מ- S אל v . מסלול פשוט זה, לא מכיל את e כיון $T \notin e$.

ב- P חייבת להיות קשת שחותча את החתך. אחרת, כל קשת שנעבור בה תשאיר אותנו בחתך, וזה לא יוכל להגיע אל הצד השני של העץ, מעבר לחתך, שבהכרח יש שם קוזקודות כיון $S \subset V \neq \emptyset$.

נסמן $e' = (u', v') \in P_1^{u \rightsquigarrow u'}$ את הקשת הראשונה P שהותה את החתך. ככלומר, $e' \rightarrow P_2^{v' \rightsquigarrow v}$

וניה $T' = (V, E_{T'}) = E_T \cup \{e\} \setminus \{e'\}$. $E_{T'} = E_T$.icut נוכיה כי T' הוא MST . ווכיה T' הוא עצם פורש: עלינו להוכיח כי הוא קשור וכן כי $|E_{T'}| = |V| - 1$, ואז בהכרח הוא עצם פורש. נשים לב כי

$$|E_{T'}| = |E_T| + 1 - 1 = |E_T|$$

כיון ש- T הוא עפ"מ, הוא בהכרח ע"פ וולקו $|E_T| = |V| - 1$, ומכוון ש- $|V| - 1$ נוכיה קשרות. יהיו $V \in y, x$. נרצה להוכיח כי קיוס מסלול C' בינו x ל- y . T הוא עפ"מ, ולכן קיוס מסלול פשוט וחוץ מ- x ל- y . נסמן את המסלול C' .

אם $P' \notin e'$, אז המסלול P' קיים גם בערך T' וכן יש מסלול בינו x ל- y . (ככלומר, הצלע שהורדנו לא נמצאת על המסלול בין השווים).

אם $P' \in e'$ (ככלומר, הצלע שהורדנו בבניית העץ נמצאת על המסלול הפשטוט), נניח בה"כ כי u' מופיע לפני u' ב- P' ונסמן את המסלול $u' \rightsquigarrow x$ ב- C'_1 ואת המסלול $y \rightsquigarrow u'$ ב- C'_2 . ככלומר,

$$P' : (P_1^{x \rightsquigarrow u'} \rightarrow e' \rightarrow P_2^{v' \rightsquigarrow y})$$

icut ניה מסלול C' מ- x ל- y נאפו הבא:

$$P'_{1x \rightarrow u'} \rightsquigarrow (P_1^R)_{u' \rightarrow u} \rightsquigarrow e_{u \rightarrow v} \rightsquigarrow (P_2^R)_{v \rightarrow v'} \rightsquigarrow (P_2')_{v' \rightarrow y}$$

הערה. P_1 הוא המסלול שMOVIL אותונו מ- u ל- v . נרצה לכתוב R (רוורס) כי נרצה לנקת CUT ב المسلול ההפוך. נזומה עכ"ר P_2^R .

סה"כ, בנינו מסלול מכל ב- T' שעובר מ- x אל y , ולכן העץ קשור, וכן T' הוא עצם פורש של G .

2. נוכיה כי $w(T') \leq w(T')$, כיון ש- $w(T')$ הוא המינימי, ואם $w(T') \leq w(T)$ הוא קטן מהמינימי וכפרט מכך. נשים לב כי

$$w(T') = w(E_T \cup \{e\} \setminus \{e'\}) = w(T) + w(e) - w(e')$$

נתו כי e קשת קלה ביותר, וכן $w(e) \leq w(e') - w$ ולו

$$w(T') = w(T) + w(e) - w(e') \leq w(T)$$

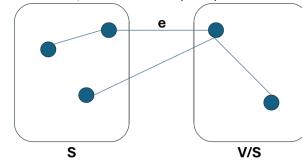
כיוון שהוורזנו משווה $w(T')$. סה"כ, $w(T') \leq w(T)$, כלומר T' עפ"י ולו המשקל שלו קטן משל כל עפ"י אחר, וכן המשקל של T' הוא הקטן ביותר. סה"כ T' הוא עץ פורש מינימום, שמכיל את הקשת e .

■

3.5 כיווץ קשותות

כרגע בסיסי מאד, בהתחשב בлемה שעמלנו קשות להוכיח עמו, נוכל למצוא את הקשת הקלה ביותר בחתך, לפי הлемה היא נמצאת ב- MST , ולהפעיל רקורסיה על צד S ורקורסיה על צד V/S וככה באופן רקורסיבי בשיטת הפרד ומושל למצוא את העץ הפורש. זה לא עובד. למה?

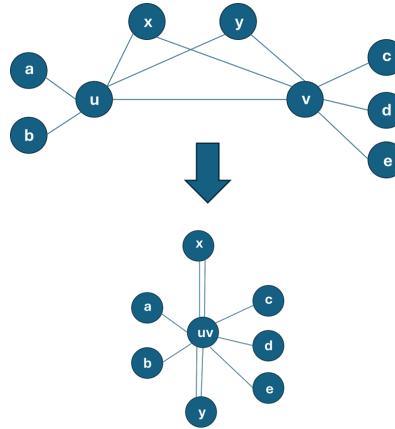
תגלו בתמונה. מזבר גורף קשור. e היא הקשת הקלה ביותר. על פיו - אלה. נעשה רקורסיה כפי שאמרנו,icut, כאשר נפעיל רקורסיה על החתך (צד S), קיבל שהגורף אליו קשור עוד. הקודקוד התיכון לא מחובר לקודקודים העליונים. וכן - זו אינה תה בעיה.



בעיה נוספת ש針טרך לטפל בה - איך נמנע ממצב בו כל הקשותות גורף בעלות אותו ערך, ו נניח, לפי שיטה זו בוחרים קשת קלה ביותר ומתקדים - נמצב זה מתקע כי כל הקשותות גורף באותו גודל.

כיווץ קשותות:

התהlik יהיה די פשוט: נניח ונרצה להעלים" את הקשת בין $v \rightarrow u$, כל שנעשה יהיה כמו בתרשים מטה: נאחסן את v , u לקודקוד משותף בשם uv , את השכנים (הלא משותפים שליהם) נחבר באמצעות הוספת קשת uv בין כל אחד מהשכנים. את השכנים המשותפים שליהם (x, y) נחבר ל- uv באמצעות הוספת שתי קשותות. כל קשת תקבל את הערך שהיא לה מקודם עם v, u .



פורמלית: יהיו מולטי גרף $G = (V, E)$ לא מכוון, פעולה כיווץ על קשת $e = (u, v)$ גראף $G/e = (V/e, E/e)$ כך ש: מיצירת גראף חדש

$$V/e = V \cup \{uv\} \setminus \{u, v\}$$

נדיר פונקציה: $F: V \rightarrow V/e$ כך ש:

$$F(x) := \begin{cases} x & x \neq v, u \\ uv & x = u \vee x = v \end{cases}$$

הפונקציה F ממחה את הקודקודים המקוריים, לקודקוד החדש שמייצג אותם. כעת נגידר את E/e :

$$E/e := \{(F(x), F(y)) | (x, y) \in E\}$$

כז למשל: אם $u = x$ וכן $v \neq y$. בהפעלה הבאה של הפונקציה נקבל $(F(x), F(y)) = (F(u), F(v)) = (uv, uv)$ כיוון שכל מה שנשלה לה כעת נמצא ב- uv .

נשים לב כי $(F(u), F(v)) = (uv, uv)$ לולאה עצמאית. עם זאת - **בגרף לא מכוון אסור לולאות עצמאיות**. לכן, בתהליך היפוי נעלמת קשת אחת - אין לולאה עצמאית, הקשת שהייתה בין u לבין v איננה עוד. כמו כן, אם נתקלנו בגרף בו יש שלוש קשות בין u לבין v - כולם נעלמות בתהליכי היפוי, לא רק אחת מהן.

3.6 אלגוריתם גנרי של MST

כעת נדוע באלגוריתם גנרי, אין לנו עניין בזמן הריצה שלו וAIN זריך אותו - כשמו כן הוא. חסרים לנו יותר מדי פרטים טכניים וחשובים לזמן הריצה ולהפעלה שלו. אם כן, הוא חשוב כרענון כללי עליו נתבסס בהמשך. להלן האלגוריתם $MST(G=(V,E), w)$:

1. $E_T = \emptyset$
2. while $|V| > 1$:
3. let e be the minimum weight edge of **some** cut in G
4. Add e to E_T
5. Contract e
6. Return E_T

נשים לב - שהאלגוריתם ממש מותבבס על מנת הבחירה החמדנית, שקיים תמיד עפ"מ חתך ב- G . ויהי $e = (u, v)$ קשת קלה בחתך $(S, V/S)$. יהי $T' = (V/e, E_{T'}, S, V/S)$ עפ"מ עבור $T = (V, E_T)$. יהי G/e עפ"מ עבור G . כלומר, T' הוא עפ"מ חדש של G/e . אולם, קח את הקשת הקלה ביוור e , תכווץ אותה מהגרף ותתקבל עפ"מ חדש של G/e . אס תחק כל פעע את הקשת זו ותחבר אותה לקבוצות הקשיות שיערו עפ"מ, אתה תקבל עפ"מ עבור G . לעומת זאת, בדיקת העצם ברקורסיה שמתבצע שוב ושוב, מה שקרה באלגוריתם שכופיע כאו מעלה, שיעור לו עז פורש מינימום).

3.7 תוכנת לת המבנה האופטימי

למה: יהי $G = (V, E)$ מולטי גראף קשור, עם פונקציית משקל $w : E \rightarrow \mathbb{R}$. יהי $(S, V/S)$ חתך ב- G . ויהי $e = (u, v)$ קשת קלה בחתך $(S, V/S)$. יהי $T' = (V/e, E_{T'}, S, V/S)$ עפ"מ עבור $T = (V, E_T)$. יהי G/e עפ"מ עבור G . לעומת זאת, T' הוא עפ"מ חדש של G/e . אס תחק כל פעע את הקשת זו ותחבר אותה לקבוצות הקשיות שיערו עפ"מ, אתה תקבל עפ"מ עבור G . לעומת זאת, בדיקת העצם ברקורסיה שמתבצע שוב ושוב, מה שקרה באלגוריתם שכופיע כאו מעלה, שיעור לו עז פורש מינימום).

הוכחה:

יהי $G = (V, E)$ מולטי גראף קשור, עם פונקציית משקל $w : E \rightarrow \mathbb{R}$. יהי $(S, V/S)$ חתך ב- G . ויהי $e = (u, v)$ קשת קלה בחתך $(S, V/S)$. יהי $T' = (V/e, E_{T'}, S, V/S)$ עפ"מ עבור G/e . יהי $E_T = E_{T'} \cup \{e\}$.

נרצה להוכיח כי T הוא עז פורש מינימום:

1. נוכחה T עז פורש: נוכחה כי T גראף קשור ללא מעגלים, ושהוא תת גראף של הגראף המקורי $(E_T \subseteq E)$. כל שינוי הוראות קשור + ללא מעגלים. נוכחה כי הוא קשור וכי $|E_T| = |V| - 1$. ראשית, נוראה כי

$$|E_T| = |E_{T'} \cup \{e\}| = |E_{T'}| + 1$$

אם כן, T' הוא עפ"מ ולכו מתקיים גם $|V| - 2 = |V| - 1 - 1 = |V| - 1 - 1 = |V| - 2$.
 $|E_{T'}| = |V| - 2$.
 $|E_T| = |V| - 1 = |V| - 2 + 1 = |V| - 1 = |E_{T'}| + 1 = |E_T|$, כנדרש.
 icut, נוכחה קשורות. יהיו $x, y \in V$. אם המסלול הפשוט $x \rightarrow y$ מ(x) $\rightarrow F(y) \rightarrow F(x)$ לא משתמש בטען קזוקוד פנימי, אזו אותו מסלול קיים גם ב- T .
 אחרת, במסלול uv כקזוקוד פנימי. החלפת הקזוקוד uv במסלול הפשוט E קשת $v \rightarrow u$, מייצרת מסלול פשוט $ux \rightarrow y$.
 מזועג T' עפ"מ G/e ובפרט קשור. לכן ש, קיים מסלול $y \rightarrow uv \rightarrow x$, כאשר מסתכל ב- T' זהה, יוכל להסתכל על אותו מסלול בדוק, בתוספת הקשת $v \rightarrow u$. כלומר $y \rightarrow v \rightarrow u \rightarrow x$, מסלול זה קיים גם ב- T כי לא שינויו דבריס פרט uv , ולכן סה"כ קיים מסלול פשוט בין x ל- y .
 לכן T קשור.

סה"כ T קשור וכו 1 – $|E_T| = |V|$ ולכו T עז פורש.
 2. נוכיח כי T הוא בעל משקל קטן ביחסו: נניח בשילוב כי T לא עז פורש מינימום. אז קיים \hat{T} עפ"מ עכור G . מלמהו, נניח בה"כ כי \hat{T} מכיל את e (הקשת הקטנה ביותר, אשר קיים עפ"מ שמכיל אותה לפי הלהמה). נכווץ את \hat{T} על e ונתקבל את \hat{T}' שמכורז:

$$\hat{T}' = (V/e, E_{\hat{T}'})$$

באשר $E_{\hat{T}'} = E_{\hat{T}}/\{e\}$
 ונשים לב כי \hat{T}' הוא עפ"מ עכור $e : G/e$
 $|E_{\hat{T}'}| = |E_{\hat{T}}| - 1 = |V| - 1 - 1 = |V/e| - 1$.
 ב. וכן, יש להראות כי $\forall x, y \in V/e$ יש מסלול $x \rightarrow_e y$:
 אם המסלול \hat{P} מ x ל y ב \hat{T} משתמש ב e אז הוא גורא לכך –

$$x \rightarrow u \rightarrow_e v \rightarrow y$$

לאחר הכיווץ על G/e מסלול זה ב \hat{T}' יראה כך: $y \rightarrow uv \rightarrow x$, כלומר מסלול זה הוא מסלול x ל y ב \hat{T}' .
 אחרת, המסלול \hat{P} לא משתמש ב e , ולאחר הכיווץ הוא ישאר אותו מסלול גזוק.
 סה"כ, ככל מקורה קיים מסלול $x \rightarrow y$ – לנו הגורף קשור.
 סה"כ \hat{T}' קשור + $|E_{\hat{T}'}| = |V/e| - 1$ ולכו \hat{T}' הוא עז פורש.
 ג. נראה כי

$$w(\hat{T}') = w(\hat{T}) - w(e) < (*)w(T) - w(e) = (***)w(T')$$

(*) כי מההנחה \hat{T} הוא עפ"מ ולכו $w(\hat{T}) < w(T)$ וכך $w(\hat{T}') < w(T')$ הוא העז
 שהוא מפומו את e .
 סה"כ, קיילו \hat{T}' הוא עפ"מ על G/e , בפרט $(\hat{T}', w, G/e)$ כתיהו! כי T' הוא עפ"מ על G/e (מהנתו), ולכו $(T', w) \leq w(\hat{T}')$ ומכורז \hat{T}' הוא עז פורש.
 מסקנה – T הוא בעל משקל קטן ביחסו, וסה"כ T הוא MST. נוצרש.

■

3.8 האלגוריתם של פרימ (Prim)

הרעיון באלגוריתם: להתמודד עם הבעיה עם איזה חתך נתחל ובוחור בכל שלב?". בReLUון זה בוחרים חתך שבו בצד אחד קודקוד אחד, ובצד השני שאר הקודקודים. כמו באלגוריתם הנפרי, נרצה לבוחר את הקשת הקלה ביותר (אם יש כמה באותו גודל – בוחרים אחת מהן). באיטרציה הראשונה – בוחרים קודקוד שריוטני. לאחר מכן ממשיכים איתנו עד הסוף, נניח שהקודקודים הינם u_n, \dots, u_1 . עם הקשותות $u_2 \rightarrow u_1$ וכו'. מוכוצחים את $u_2 \rightarrow u_1$.
 כעת מקבלים u_1u_2 מצד אחד, ומಹatz השני של החתך שאר הקודקודים u_n, \dots, u_3 וכו'!
 ממשיכים את האלגוריתם עד שמקבלים קודקוד יחיד $u_1u_2u_3\dots u_n$.
 האלגוריתם הולך להשתמש בתור קדימיות, כאשר יש לו שתי פעולות עיקריות: *Init*, *ExtractMin*

האלגוריתם :

$$(G = (V, E), w) \quad E_T = \emptyset .1$$

2. עבור כל $v \in V$ בצע:

א. $u.key = \infty$ (המפתח יגיד את משקל הקשת **הקללה** ביוטר בחותך שנוגעת לו וועברת דרך החותך לצד השני - נשים לב: מחליפים את הערך של המפתח רק אם הערך קטן יותר מהערך הקודם שלו.).

ב. $u.\pi = null$ (הפאי יגיד לנו מי הקודקוד הצד השני של הקשת, שהמשקל שלו הוא u - ושוב, נשים לב שערך הפאי השתנה רק אם ערך key השתנה, וישתנה למי שמחזיק בערך זה.).

3. בחר קודקוד שירוטי $r \in V$ ואתחל $r.key = 0$ ו- $Q.init(V)$.4

5. כל עוד $|Q| \geq 1$ בצע:

א. $Q = Q.extract.Min()$ (סקול לבחירת קשת קללה ביוטר" - מוציאים אותו מהטור)

ב. אם $u \neq null \rightarrow E_T$ (חווב לשים לב - בפעם הראשונה שנכנס לשלב 5 באלגוריתם, נכנס בהכרח במצב בו $u = null$, כיון שבשלב 2' אתחלו את כלום להיות $null$, ולכן לא נוציא כלום).

ג. עבור כל $u \in ADJ[u]$: (תעביר על השכנים של u)

אם $u \in Q$ ואם עדין בתוך התור) וגם $u.\pi \neq v$

$$v.key = w(u, v) .1$$

$$v.\pi = u .2$$

6. החזר E_T .

כוננות האלגוריתם: נובעת מנכונות האלגוריתם הגנרי. בכל שלב מסמלצים"cioyz על הקשת שבחרנו, ובכל שלב מסתכלים על החותך כקודקוד שבחרנו כתע עם כל מה שקדם, למול מה שנשאר. זה אלגוריתם שמאוד דומה לאלגוריתם הגנרי, ומשם נכונתו.

זמן הריצה:

ניתן לראות שלבים 3 – 1

-
-
-

 $O(|V|)$ זמן, שכן עוברים על כל הקודקודים.

שלב 4 – תלוי ב- $Queue$.

שלב 5' – גם כן, תלוי בסוג $Queue$. אבל, כמה פעמים נבצע הוצאה? $O(|V|)$ הוצאות, כי כל קודקוד יכול לצאת פעם אחת.

נשים לב, שככל קודקוד יכול לצאת מהטור ככל היותר פעם אחת. כאשר קודקוד יוצא מ- Q , האלגוריתם עובר על כל השכנים שלו, ובמקרה הגרוע כל שכן גורר עדכון מפתח אחד. לכן, שלב 5' יעלה $deg(u)$. אבל – כמה פעמים בכלל הם יכולים להקטין את המפתח שלהם (key , להכרח להקטין לפי מה שהסבירו?) מס' הפעמים הוא $\sum_{u \in V} deg(u) = 2|E| = O(|E|)$

لسיכום – זמן הריצה תלוי ב:

א. **מערך:** אתחול עליה $O(|V|)$, ויתבצע פעם אחת. הוצאת המינימום עולה $O(|V|)$, ויתבצע $|V|$ פעמים – וסה"כ עליה $O(|V|^2)$, הקטנת מפתח תעלה $O(1)$ ויתבצע $2|E|$ פעמים. סה"כ סיבוכיות הזמן במערך תהיה $O(|V|^2 + |V| + |E|)$, כמו כן $|V|^2 \leq |E|$ ולכן סה"כ סיבוכיות זמן הריצה $O(|V|^2)$.

ב. עירמה ביןארית/בינויית: אתחול עלה $O(V)$ ויתבצע פעמיים, הוצאה המינימום עלה $O(\log|V|)$ ויתבצע $|V|$ פעמיים, וכן הקטנת מפתח עלה $O(\log|V|)$ ויתבצע $2|E|$ פעמיים. סה"כ סיבוכיות הזמן בעירמה תהיה $O(|V|\log|V| + |V| + |E|\log|V|)$ וכן, כמובן, שהרף קשרים מתקיים $|E| \geq |V| - 1$ בפרט $|E|\log|V| \geq |V|\log|V|$ ולכן סה"כ סיבוכיות זמן הריצה הינה $O(|E|\log|V|)$.

ג. עירמת פיבונאצ'י: אתחול ב($|V|, O(|V|\log|V|)$, הוצאה מינימום $|V|$ פעמים שתעלת \log , וכן הפתחת מפתח עלה $O(1)$ לשינוין שתתבצע $2|E|$ פעמיים. לכן סה"כ סיבוכיות הזמן בעירמת פיבונאצ'י ז閻ן ריצה ($O(|V|\log|V| + |E|)$).

לא מומלץ להשתמש במערך. נshallת השאלה מה עדיף – בעירמה ביןארית או בעירמת פיבונאצ'י? תמיד מתקיים הר依 כי $|V| \geq |E|$, ולכן ניתן לראות שעדיף להשתמש בעירמת פיבונאצ'י ז閻ן ריצה של $O(|V|\log|V| + |E|)$.

3.9 האלגוריתם של ק魯סקל

הרעיון: נבחר בכל פעם את הקשת הקללה ביותר **בכל הגרף**, ונוסף אותה לעץ פורש המינימום. (בפרט, היא תהיה הכללה בחתך כלשהו).
הקשה – לדאגן שאין מעגלים / אין לולאה עצמית בזמן הcyoz.
להלן האלגוריתם:

MST-KRUSKAL($G = (V, E), w$):

1. $E_T = \emptyset$
2. for each $u \in V$:
 - a. make_set(u)
 3. for every edge $e = (u, v) \in E$ in increasing order of weights
 4. if $\text{find_set}(u) \neq \text{find_set}(v)$
 - a. Add (u, v) to E_T
 - b. union (u, v)
 5. return E_T

כפי שניתן לראות – האלגוריתם משתמש בינוין פיינד. בתחילתו, לכל קודקוד ניצור קבוצה. לאחר מכן, נתחיל מהקשת הקללה ביותר, ונעה בסדר עולה של משקלים, כך נעבור על כל הקשתות. בכל שלב, נבדוק האם שני הקודודים שמרכיבים את הקשת נמצאים באותה קבוצה. אם לא – נוסיף את הקשת בינויהם לקבוצת הקשתות, ונאחד בין הקבוצות שלهما (נשים לב שיתכן ונוצר מצב בו הקבוצות כרגע הם $\{w, x\}, \{w, y\}$ ואנו נדרשים לבדוק את הקשת wx . אין בהם קרע קשת ולכן אנחנו נוסיף אותה ל- MST וכן נאחד בין הקבוצות לקבוצה גדולה $\{x, w, y\}$).

למעשה – מה שהאלגוריתם עושה קורה בשלבים 4 – 3. אם שני האיברים זרים זה זה ולא נמצאים באותה קבוצה, חייבים להוסיף קשת שתחבר ביניהם בעז, בשביל שהיא עצה פורש ובפרט קשר. אם הם כבר באותה קבוצה, אין צורך להוסיף קשת שתחבר ביניהם. מהיכן מגיע המינימום? מההמעבר על הקשתות לפי הקשת הנמוכה ביותר עד לגודלה ביותר. בכל מקרה, נעבור על כל הקשתות – אך כשנגייע למצב שכל האיברים באותו קבוצה ויש קבוצה אחת – סימנו.

כיצד האלגוריתם בודאות לא יבחר מעגל? נניח ובחרנו קודקודים u_k, u_1, \dots, u_n . נרצה לעמוד על קשת $u_k \rightarrow u_1$. אם נוסיף אותה, בהכרח יוצר מעגל. בשלב 4 אנחנו בבדיקה

בודקים את זה - לאחר שלב 4 נקבל תשובה שהקודדים באוטה קבוצה ולכז לא נוסיף קשת זאת, וכך לא יוצר מעגל.

זמן הריצה:

לפי האלגוריתם ניתן לראות כי מבצעים ($O(|V|)$ פעולות *makeSet*, כלומר $O(1)$ וסה"כ $O(|V|)$, וכן מבצעים $1 - |V|$ פעולות *union*, שככל אחת עולה $O(\log^*|V|)$ ולכז סה"כ $|V|(\alpha|V|)$). כמו כן מבצעים $|E|$ פעמים *findSet* שעולה $O(\alpha|V|)$ ולכז סה"כ $O(|E|\alpha|V|)$. כמו כן, עלינו למיין את הקשתות E , לכאורה ניתן למיין ב- $|E|\log|E|$ אך עם מעט יותר מידע על סוג המשקלים ניתן גם ב- $O(|E|)$ זמן. סה"כ סיבוכיות זמן הריצה תהיה -

$$O(|V| + (|E| + |V|)\alpha|V| + \text{sort}(E)) = O(|E|(\alpha|V|) + \text{sort}(E))$$

אם $|E| = |E|$ אז סיבוכיות זמן הריצה הינה $O(|E|(\alpha|V|) + |E|)$, אחרת $(O(|E|(\alpha|V|) + |E|\log|E|)$ בהשוואה בין זמני הריצה של פרימס וקרוסקל - בכךן כל קروسקל ינצח. אך אם מס' הקשתות גדול יחסית, ממש גדול יחסית - אז עדיף להשתמש בשל פרימס. אחרת, של פרוסקל ינצח.

3.10. תכונת המעלגים הכבדים של MST (טרגול)

למה 1. יהיו $G = (V, E)$ גרף לא מכון עם פונקציית משקל על הקשות $\rightarrow : E \rightarrow \mathbb{R}$. יהיו C מעגל ב- G כך ש- $e \in C$ היא קשת כבדה בمعالג. אז, קיים MST שלא מכיל את e .

הוכחה: נניח בsvilleה שקיים MST שמכיל את $e = (x, y)$. יהיו T כטיל אשר $e \in T$. נבנה בגרף שמתפרק מהסרת הקשת e מ- T . ככלומר $T \setminus \{e\}$. מתקבל גראף עם $|V| - 2$ קשותות (קודם לכן היה $|V| - 1$ כי T הינו עצם) ולכז הוא בהכרח אינו קשרי. יתרה מזאת, הקודדים x ו- y נמצאים ברכבי קשר שונים בגרף זה. נסמן את רכיב הקשירות שמכיל את x בתו S ונסתכל על החטך $(S, V \setminus S)$. אנו יודעים כי המمعالג C מכיל מסלול mx לעבר G ללא הקשת e , ככלומר קיימות קשתות $C \setminus \{e\}$ ב- $e' = (u, v)$ שחוויה את החטך $(S, V \setminus S)$. בה"כ נניח כי $v \in S$ ו- $u \in V \setminus S$. נביט בגרף G . נרצה לטען עז. ב' T' ישנו בדיקת $|V| - 1$ קשותות ולכז נטען כי T' קשרי וזה יהיה מספיק.

יהיו $a, b \in V$ קודדים. אם שניהם באוטו צד של החטך אז יש בניהם מסלול כי T קשרי. אחרת נניח שהם בצדדים שונים של החטך, בה"כ $a \in S, b \in V \setminus S$ או $a \in V \setminus S, b \in S$. במקרה הראשון נוכיח כי $a \sim u \sim v \sim b$ באשר מובטח לנו שיש מסלול ma לעבר m ומשם lb כי כל צד של החטך קשרי. אם כן הוכחנו כי T' הקשרי. נרצה לראות שמשקלנו קטן או שווה משקלו של T ב כדי לקבל סתירה (כי T עפ"מ ובפרט משקלו מינימלי).

$$w(T') = w(T) + w(e') - w(e) \leq w(T)$$

כיוון ש- e הייתה קשת כבדה בمعالג C והקשת e' מקיימת $w(e') \leq w(e)$. סה"כ סתירה לכך שהוא MST במשקל מינימלי.

מדוע אנחנו זוקרים לлемה זו? מכאן עולה רעיון אלגוריתמי אלטנטיבי לריעון שראינו בהרצאה אודות MST. יהיו מעגל C ו- G , אזי יודעים שקיים עפ"מ ללא קשת כבדה ביותר ב- C , שכן ניתן להוריד את הקשת הזו מהגרף ולהמשיך ברקורסיה על הקשתות שנותרו. קروسקל, הצע במאמרו

המקורי גם את האלגוריתם הבא. שנקרא גם "אלגוריתם מחיקה כפולה".

- :*reserve – delete Algo*($G = (V, E)$, w)

 - .א. מין את E בסדר יורד, כי הסדר לאחר המין: $e_1, \dots, e_{|E|}$
 - .ב. לכל $i = |E| \dots 1$ עד i מחק את e_i .
 - .ג. אם הגרף ללא e_i אינו קשיר, החזר את e_i לgraf.
 - .ד. החזר את קבוצת הקשתות שלא נמחקו בmphלך ריצת האלגוריתם.

ונוכיח את האלגוריתם באמצעות למה 1 ובעמצעות הלמה הבאה:
למה 2. תהי $F \subseteq E$ קבוצת הקשתות שנשארה בגרף בסוף הולאה של האלגוריתם $reserve$ –
או כיון עפ"מ $T = (V, ET)$ של G באשר $ET \subseteq F$.
הוכחה: ההוכחה תהיה באינדוקציה על מס' האיטרציות של הולאה.

בבסיס: קל לראות כי לפנין תחילת הוללה, הטענה נכונה באפון ריק. קבוצת הקשתות $F = E \subseteq E$ ממש ואכן קיימים עפ"מ T של G שצלעויות מוכלות ב- E' , שכן קיימים עפ"מ בכל גראף קשריר.

יעד: נניח שהטענה נכונה עבור קבוצת קשתות $F \subseteq E$ בסוף איטרציה מסוימת של הוללה, כלומר קיימים עפ"מ (V, E_T) של G באשר $T = E_T \subseteq F$. נסתכל על קבוצת הקשתות $F' \subseteq E$ שמתאפשרת בסיום האיטרציה הבאה של הוללה ונרצה להוכיח כי קיימים עפ"מ $T' = (V, E_{T'})$ של G כך ש- $E_{T'} \subset F'$. נחלה למקרים:

א. אם לא הוסרה קשת, אז בהכרח $F' = F$ ולפי הנחת האינדוקציה הטענה מתקיימת.
 ב. הוסרה קשת $e \in F$, מהגדרת האלגוריתם הקשת e הייתה חלק מעגל C ב- F , נראה כי כל קשת אחרת במעגל לא נבחנה עד כה במהלך ריצת האלגוריתם אחרת האלגוריתם היה מסיר אותה (ב証明 סידרנו את קשותות הגרף לפי גודלו, ובודאות תופע קודם כל הקשת הכבודה ביותר במעגל).
 שכן הגרף שמושירה ישר קשור לאחר הרות קשת יחידה מהמעגל C . מכאן נקבל כי e היא קשת כבדה במעגל. מכאן, לפי lemma 1, בהכרח קיימים MST של G' שלא מכיל את e , בשילוב עם הנחת האינדוקציה שאומרת ש $up'(G')$ הוא גם $up'(G)$ לנכל להסיק כי $up'(M) = (V, E_{T'})$.

3.11 השפעת סדר מינו הקשיות על הפלט בהערכת האלגוריתם של קרווסקל

לכל $U \subseteq M$ (V, E_U) קיימים סדר של E שנסמננו π שהוא מינו של הקשיות $U \subseteq F$.
 מושג זה מושכלן בסדר עולה, כך שההרכזה של האלגוריתם של קורוסקל על G בהתאם ל- π תחזיר את T .

הערה חשובה. כאשר הוכחנו את הטענה הסתכלנו על עפ"מ כלשהו T של G והראינו עבורי סדר מינו כך שאלגוריתם קורוסקל מוציא את T כפלט. טוועות נפוצה בשאלות מסווג זה היא נסיוון להסתכל על עפ"מ שהוא פלט של אלגוריתם קורוסקל (או כל אלגוריתם שפותר את הבעיה לצורך העניין) ונסיוון לשלוטן טענות לבבו. שימוש ביבי עפ"מ של גוף הוא אובייקט מותמעי של הגרא, ויכולים להיות כמה עפ"מים שונים. אלגוריתמים כמו פריטים קורוסקל וכובע הם תהליכי חישוביים שמוצאים אובייקט מותמעי שכחיה (bijuliot), אך הוא ספציפי מכיוון שהוא אפשרי, ולכן כאשר מתבקשים להוכיח טענה על אובייקט מותמעי שירותוי אסור לנו להניח שהוא פלט של אלגוריתם זהה או אחר (זהו מקרה פרטני של דגש הדודרשוں מכם לשים לב כי ישן שאלות בהן הטענה היא טענת לכל ולא טענת קיים, וכן להיפך).

לפי הטענה אם ידוע כי G מכיל קשתות ממושכלים שונים, בהכרח קיים סדר מיוון יחיד ולכזו קיימים MST יחיד!

הוכחה. יהי T עפ"ם של G , נגדיר את π באופן הבא: π יהיה מינון חוקי בסדר עולה של E , כאשר אם ישן קשתות במשקל שווה ניתן עדיפות במינון לקשותות שנמצאות ב- T .
כעת נרצה להוכיח שריצת האלגוריתם של קروسקל על קשתות בחתams G כפי שהוגדר לעילו, מחזירה את T כפלו:

נב"ש שלא, אויה האלגוריתם של קروسקל מוציא כפלט עפ"ם אחר \hat{T} , כך ש: $\hat{T} \neq T$.
נביט בקשת הראשונה (עפ"פ סדר המינון π_T) (u, v) כך ש $e = \hat{T}, e \notin T$:
 $P_T = e_1, e_2, \dots, e_k$ מסלול היחיד ב- T מ- u ל- v . נסמן: e_1, e_2, \dots, e_k, e נשים לב כי לפחות אחת מהקשותות ב- P_T לא נמצאת ב- \hat{T} (אחרת קיים מעגל: $e_i \notin \hat{T}, e_i \in P_T \subseteq T$, בסתרה להיות עז). נסמן את הקשת הזו: (x, y) , וכן את הקשת e_i .

הבחנה: $w(e) < w(e_i)$
הוכחה: נב"ש $w(e) \geq w(e_i)$. לפי הגדרת π_T , e_i מופיעה לפני e ב- π (מפני ש- $e_i \notin \hat{T}, e_i \in T$).
וגם $w(e) \geq w(e_i)$. מכיוון ש- $e_i \notin \hat{T}$, נסיק שאשר האלגוריתם של קروسקל בחר את e_i, e_i סורה מעל עם הקשתות שנבחרו לפני \hat{T} , ואויהן הקשתות נבחרו ל- \hat{T} לפני ש- e נבחרה ל- \hat{T} . ולכן כל הקשתות האלה נמצאות ב- T (לפי הגדרת e). מכאן שקיים מעגל ב- T (שמכיל את הקשת e_i) בסתרה לכך ש- T עז.

כעת, נבנה מ- T עז אחר $T' = (V, E_{T'}) = (V, E_T)$ שמשקלו קטן יותר וו תהייה סתירה לכך ש: T' עפ"ם של T .
נגדיר: $E_{T'} = (E_T \setminus \{e_i\}) \cup \{e\}$

טענה: T' הוא עז.
ברור כי $|E_{T'}| = |E_T| - 1 = |V|$ ולכן מספיק שנראה כי תוכנות הקשיות מתקיימת ב- T' .
נראה כי קיים מסלול ב- T' בין כל זוג קודודים בעז:
יהיו $s, t \in V$ בغالל ש- T עז אנחנו יודעים שקיים מסלול יחיד בין s, t ב- T . נחלק למקרים:

מקרה 1: המסלול בין s ל- t ב- T לא משתמש בקשת e_i .
אויה אותו מסלול קיים בעז T' .

מקרה 2: המסלול בין s ל- t ב- T משתמש בקשת e_i .
נניח כי המסלול הוא מהצורה הבאה: $P_{s,t} = s \rightsquigarrow_{P_{s,x}} x \rightarrow y \rightsquigarrow_{P_{y,t}} t$.
תחילה נשים לב כי המסלולים $P_{s,x}, P_{y,t} \subseteq T'$ מפני שאינם משתמשים בקשת e_i .
כעת, נזכיר שקיים ב- T מסלול מ- u ל- v , P_T שהקשת e_i נמצאת בו, אויה ניתן לפצל אותו לשולש מסלולים:

החלק $P_{u,x}$ החולק בمسلול P_T מהקודוקוד u עד לקודוקוד x .
הקשת $(x, y) = e_i$.
החלק $P_{y,v}$ החולק בمسلול P_T מהקודוקוד y עד לקודוקוד v .
נשים לב כי $P_{u,x}, P_{y,v} \subseteq T'$ וגם $P_{u,x}, P_{y,v} \subseteq T$ וולכן $e_i \notin P_{u,x}, P_{y,v}$ ולכן נבנה מסלול ב- T' שלא משתמש בקשת e_i בaczורה הבאה:
 $P'_{s,t} = s \rightsquigarrow_{P_{s,x}} x \rightsquigarrow_{P_{x,u}} u \rightarrow v \rightsquigarrow_{P_{v,y}} y \rightsquigarrow_{P_{y,t}} t$

בזה"כ הראינו כי קיים מסלול בין כל זוג קודודים ב- T' וגם כי $|E_{T'}| = |V| - 1$ ולכן T' הוא עז.

כעת כאשר הוכחנו כי T' הוא עז נראה סתירה למינימליות של T :

מההבחנה אנחנו יודעים ש- $w(e) < w(e_i)$.
ולכן נקבל כי: $w(T') = w(T) + (w(e) - w(e_i)) < w(T)$.
בסתירה להיות עז T עפ"ם.

□

4 הריצאות $shorts path - SSSP - 3 + 4$

כיצד מודדים מהו המסלול קצר ביותר?

אם הגרף אינו ממושקל: עלות המסלול היא מס' הקשתות במסלול = אורך המסלול.
 אם הגרף כן ממושקל: עלות של מסלול = סכום משקלן הקשות על המסלול.

הגדרה: עבור $v \in V$, u נסמן את העלות המינימלית של מסלול מה u ל v ב $\delta(u, v)$.
 יהיה $(u, v_1, \dots, v_{k-2}, v) = P$ המסלול קצר ביותר בין u ל v (אם קיים).
 אם G ממושקל:

$$\delta(u, v) = \sum_{v \in P} w(v)$$

אם G אינו ממושקל:

$$\delta(u, v) = |P| = k - 1$$

אם לא קיים מסלול בין u ל v נגדיר:

$$\delta(u, v) = \infty$$

הגדרה: מסלול מה u שעלוותו היא $\delta(u, v)$ יקרא מסלול קצר ביותר.

הערה: יתכו שכוו זוג קודקודים יש יותר ממסלול אחד קצר ביותר.

הערה: רוכ האלגוריתמים שנראה בהרואה יהיו עבר גוף מכובו. כיצד זה פותר את הבעיה עבור גוף שאינו מכובו? אם האלגוריתם יוזע לפטור את הבעיה עבר גוף מכובו, יוכל להפир כל גוף לא מכובו למולטי גוף מכובו: כך שככל קש תיב $a \longleftrightarrow b$ ותתרוגש לשתי קשותות $a \rightarrow b, b \rightarrow a$.

הערה: ישם מקרים בהם יש אלגוריתם יותר מהיר עבר גוף לא מכובו.
 הערה: ניתן להשתמש בפתרון עבר המקרה הממושך למקרה הלא ממושך, אם נזריר פונקציית משקל קבועה. למשל כל הקשותות בעלות אחת.

4.1 בעיית מציאת המסלול קצר ביותר

לבעיה זו יש מס' גרסאות. נשים לב כי הן מדורגות בעות מהקלה לקשה.

1. זוג קודקודים -

קלט: $G = (V, E)$ זוג קודקודים $u, v \in V$.
 פלט: לחשב את $\delta(u, v)$ ואולי אף למצוא מסלול מה u שהוא קצר ביותר.

2. מקור יחיד - (Single Source Shortest Paths (SSSP))

קלט: $G = (V, E)$ וקודקוד $s \in V$ שיקרא מקורו.
 הפלט: לחשב עבר כל $v \in V$ את $\delta(s, v)$ ואולי גם למצוא מסלול קצר ביותר מס' לכל $v \in V$.

3. כל הזוגות - (All Pairs Shortest Paths (APSP))

קלט: $G = (V, E)$
 פלט: לכל V , $u, v \in V$ להחזיר את $\delta(u, v)$ ואולי אף למצוא את המסלול קצר ביותר לכל $u, v \in V$.

הערה: בעיה 1 מוכלת בתוך בעיה 2, ועם זאת כדי שיראה בהמשך לא קיים אלגוריתם שפותר אותה יותר טוב מעת בעיה 2. לעומת עדרון אלגוריתם יעיל יותר בסיבוכיות עכבר בעיה 1. כמו כן, בעיה 2 מוכלת בעיה 3 - אך כו זמנו הוריצה של בעיה 2 טוב יותר משל 3.

4.1.1 איך נראה פתרון בכלל אחד מהגרסאות כפתרונות את המסלול?

זוג יחיד: מסלול $(u, v_0, \dots, v_{k-1}, v)$, שיעלה $O(|V|)$ זכרו.
מקור יחיד: נאיבטי, אפשר להחזיר $|V|$ מסלולים שונים, אחד עבור כל קודקוד מטרה. ככלומר:

$$p_1 = (s, \dots, v_1)$$

$$p_2 = (s, \dots, v_2)$$

..

$$p_n = (s, \dots, v_n)$$

מה עלות הזכרון בפתרון זה? $\sum_{i=1}^n |P_i| \leq |V| \times |V| \times \max\{|P_i|\} \leq |V|^2$.
 כתה נראה שישנה אפשרות להציג את הפתרון בצורה שתשתמש לפחות מוקם. לשם כך נשתמש בлемה החשובה מאוד הבאה -

лемה 1: תת מסלול של מסלול קצר ביותר הוא גם מסלול קצר ביותר. ככלומר, היה מסלול P ($v, w_1, \dots, w_r, x, \dots, y, z_1, \dots, z_t, u$) קצר ביותר. אז, המסלול בין x לבין y شامل במסלול P הוא גם קצר ביותר.

הוכחה: היה המסלול הקצר ביותר בין הנקודות x ו- y : $P = (v, \dots, x, p_1, \dots, p_k, y, \dots, u)$. נניח כי המסלול בין x לבין y אינו הקצר ביותר. כלומר, קיים מסלול $P_{xy2} = (x, p_1, \dots, p_k, y, \dots, u)$ בין x לבין y כך ש- $|P_{xy}| < |P_{xy2}|$. אז, ניתן על המסלול $P' = (v, w_1, \dots, w_r, P_{xy2}, z_1, \dots, z_t, u)$

$$|P'| = |P| - |P_{xy}| + |P_{xy2}| < |P|$$

בסתוריה לכך ש- P' היה המסלול הקצר ביותר בין x ו- y .

כעת, נזכיר לדון בשמרית הזיכרון: בעת שמירת מסלול כלשהו, למשל $s \rightarrow v_4 = (s, v_2, v_{14}, v_3, v_{90}, v_4)$, אנחנו שומרים מסלולים קצרים ביוטר נוספים: בין $v_90 \rightarrow v_2$ למשל, לפי הלמה שהוכחה לעיל גם הוא מסלול קצר ביותר.

כמו כן, נשים לב כי נוכל לקבל למשל שני מסלולים: $(s, v_{10}, v_5, v_8, v_4, v_9), (s, v_{10}, v_5, v_6, v_2)$ ולשרשר אותם למסלול יחיד כך:

$$(s, v_{10}, v_{5 \rightarrow v_6, v_2}^{v_8, v_4, v_9})$$

כלומר, לצורך צורה של עץ ששורשו. סה"כ זו תחיה הטכניקה - **ניתן לייצג מסלולים קצרים ביותר מ- S לכל שאר הקודקודים בגרף באמצעות עץ מסלולים קצרים ביותר.**

נשים לב - העץ לא מכיל את כל המסלולים הקצרים ביותר מ- s בגרף, כלומר: לכל $V \in s$ יהיה קיים מסלול קצר יותר שיוצג בעץ $\rightarrow s$, אך יתכן שקיים שני מסלולים כאלו קצרים ביותר באותו משקל והעץ יבחר אחד מהם בדיק שיפועו.

מסקנה: יתכן שישנם כמה עצי מסלולים קצרים ביותר.

לסיום - בגרסת מקור ייחד אנחנו נזכיר **עץ מסלולים קצרים ביותר** שעלהו תהיה בגודל M^* הקודקודים בו, $(|V|)O$. נשים לב - קודקוד שהוא ראש העץ יהיה בללא עצמיה עם עצמו, קודקודים שאין אליהם מסלול יסומנו *null*.

כל הזוגות (APSP): במצב זה נרצה להחזיר מטריצה A בגודל $|V| \times |V|$, כشنרצה להחזיר את $\delta(v, u)$ אנחנו נייצג זאת במטריצה ע"י $A_{ij} = \delta(v_i, v_j)$, סה"כ עלה $O(|V|^2)$ ארכון. אם נהיה מעוניינים במסלולים - נרצה $|V|$ עצי מסלולים קצרים ביותר, וסה"כ $O(|V|^2)$ מקום.

4.2 אלגוריתם SSSP – BFS במקורה הלא ממושקל

BFS היא סריקה לרוחב של העץ לפי רמות, מבצעים אותה באמצעות תור קדימות כפי שראינו בקורס מבני נתונים. האלגוריתם סורק את הנקודות בסדר שנקבע על פי מרחקם מהנקודת ההתחלה. **אלגוריתם זה מטפל ב-SSSP במקורה הלא ממושקל.** וכן, אורך המסלול נספר לפי מס' הקששות.

האלגוריתם מחסן 3 סוגי מידע לכל קודקוד:

1. $d[u]$ - אמצען לגביו (u, δ) . בסיום הריצה יתקיים $d[s] = \delta(s, u)$.
2. $\pi[u]$ - כלי לחישוב המסלולים הקצרים ביותר מ- s . בסיום הריצה הוא מצביע לאבא של u בעץ המסלולים הקצרים.
3. $Color[u]$ - מציין מזאה:
א. $w(hite)$ = האלגוריתם עוד לא הגיע אליו.
ב. $g(ray)$ = האלגוריתם ביקר בש ולא טיפול בו.
ג. $b(lack)$ = האלגוריתם סיים לטפל בו.

4.2.1 האלגוריתם $BFS = (G = (V, E), s)$

```

BFS( $G = (V, E), s$ )
1   for each  $u \in V$ 
2        $d[u] \leftarrow \infty$ 
3        $\pi[u] \leftarrow \text{NULL}$ 
4        $color[u] \leftarrow w$ 
5    $d[s] \leftarrow 0$ 
6    $color[s] \leftarrow g$ 
7    $Q.Enqueue(s)$ 
8   while  $Q \neq \emptyset$ 
9        $u \leftarrow Q.Dequeue()$ 
10      for each  $v \in ADJ[u]$ 
11          if  $color[v] = w$ 
12               $color[v] \leftarrow g$ 
13               $d[v] \leftarrow d[u] + 1$ 
14               $\pi[v] \leftarrow u$ 
15               $Q.Enqueue(v)$ 
16       $color[u] \leftarrow b$ 

```

האלגוריתם מתחילה בתחילת הדף $d[u]$ להיות אינסוף, את π להיות null ואת כל הצבים להיות w לא בירק. לאחר מכן נתחלה את s . נשים לב כי $d[s] = 0$ כיון שאורך המסלול הקצר ביותר מריצמו הוא אפס. תהליך האתחול מתרחש עד לשלב 7. עת כל התרור לא ריק אנחנו מבצעים: אנו מושתמשים בתורה FIFO, ומוכנסים אליו s . מעתה כל התרור לא ריק אנחנו מבצעים: מוציאים מהתרור את האיבר הראשון, עוברים על כל השכנים של הקודקוד שהוצאנו, אם הצבע שלהם belum משמע לא בירנו אותו עוד, נסמן את הקודקודים באפור, נגדיר את d שלהם להיות $d+1$ של הקודקוד שהוצאנו (שהיה השכן שלהם) וודח אחד - כי יש קשת נוספת למסלול, וכן נגדיר את אבא של כל הקודקודים האלה להיות s (הקודקוד שהוצאנו). לבסוף נכניס את כל השכנים הללו לתור. בסיום, נגדיר את הצבע של הקודקוד שהוצאנו כל סימנו לטפל בו. וכעת, עברו לטפל בקודקוד הבא בתורה. כך עד שהתרור יתרכז.

זמן הריצה: האתחול עלה $O(|V|)$ זמן, לאחר מכן מבצעים לולאה - נשים לב כי במהלך הלולאה אף קודקוד לא נצבע בלבוי, ולכן כל קודקוד נכנס ויוצא מהתרור לכל היתר פעם אחת. וכך, הפערות מותבצעות פעמיים אחת לכל קודקוד. ומכאן, שכל הלולאה של הלולאה *while* של מושתמשת *while* לכל היותר $O(|V|)$ פעמיים. באשר לעלות לולאת *for* על קודקוד s היא $O(\deg(u))$, ולכן סה"כ זמן הריצה יהיה

$$|V| + \sum_{u \in U} \deg(u) = |V| + 2|E| = O(|V| + |E|)$$

וקיילנו זמן לינארי. כמו כן נשים לב כי אסור להניח $|E| \leq |V|$, אנחנו מדברים על גרף כללי G .

נשים לב: π מגדיר את עץ המסלולים הקצרים, ככלمر ריצת BFS יכולה גם להחזיר לנו את עץ המסלולים הקצרים ביותר. באמצעות ערך π ניתן לבנות את עץ זה.

הערה: $O(|V| + |E|)$ הוא חסם תחתון לגודל הקלט ולכון אלגוריתם BFS הוא האופטימלי לפתרון הבעיה.

4.2.2 נסונות של BFS

המטרה היא להוכיח שבסוף ריצת $BFS(G, s)$ מתקיים כי $\delta(u, s) = d[u]$ $\forall u \in V$

лемה 2: למות אי שוויון המשולש. יהיו $G = (V, E)$ לא ממושקל, ויהי $s \in V$, ויהי $(u, v) \in E$. אזי

$$\delta(s, v) \leq \delta(s, u) + 1$$

כלומר, בהינתן המסלול $v \rightarrow u \rightarrow \dots \rightarrow s$, המסלול הקצר ביותר לעבר מ- s אל v חסום במסלול הקצר ביותר לעבר מ- s אל u ועוד מעבר על הקשת (u, v) , נשים לב שהוא אפשרות למסלול יתכן שיש מסלול טוב יותר קוצר יותר. מדברים על חסם בלבד!).

הוכחה: נחלק למקרים.

א. אם אין מסלול בין s ל- v : אז בכרה $1 + \infty \leq \delta(s, v)$, כנדרש.

ב. אם יש מסלול בין s ל- v : זה גורר שמתיקיות מסלול בין s ל- v : המסלול בין s ל- v בתוספת הקשת (v, u) . במקרה זה, עלות של המסלול הקצר ביותר מ- s ל- v לא יכול להיות גדול יותר מעלה המסלול מ- s ל- u). במקרה שאותה הש�ת (v, u) כיוון שאחד המסלולים האפשריים מ- s ל- v הוא מסלול שכינויו C , וכן המסלול הקצר ביותר בווזאות והיה או זה, או באורך קטן מזו. מסקנה $\delta(s, v) \leq \delta(s, u) + 1$.

лемה 3: לאחר הריצת $BFS(G, s)$ לכל $v \in V$ מתקיים $d[v] \geq \delta(s, v)$.

הוכחה: נצע אינדוקציה על מס' פעולות enqueue שיתבצעו במהלך הפעולה. נסומים.

בסיס: $n = 1$, עבור s מתקיים $d[s] = 0$ ועבור שאר הקזוקוזים u מתקיים $d[u] = \infty$ ואנו מתקיים הטענה.

צע: נניח שהטענה נכונה עבור $n - 1$ פעולות הכנסה. נוכיח לה. שינוי הערך $d[v]$ יכול להתבצע רק ממהלך מעבר על השכנים של v שעבעס בעת לנו. אם כך, יספיק להוכיח שערך כל שכנו של v , u , שבעעו לנו מתקיים $d[u] \leq \delta(s, v)$ והוא s קזוקוז כיל. לאחר העדכו מתקיים -

$$d[v] = d[u] + 1 \geq \delta(s, u) + 1 \geq (\ast)\delta(s, v)$$

כאשר (\ast) נכון לפחות אי שוויון המשולש והנחה האינדוקציה.

лемה 4: בכל אמצעי אולוגורייטס אם $(v_1, v_2, \dots, v_r) = Q$ מתקיימות שתי תכונות:

א. $d[v_1] \leq d[v_2] \leq \dots \leq d[v_r]$

ב. $d[v_r] \leq 1 + d[v_1]$ (כלומר לא יוכל שבתו בו זמניות ישנים יותר מאשר שכבות נמוכניות).

הוכחה:

נכיה את הטענה באינדוקציה על סדרת פעולות enqueue, dequeue.

בסיס: תוך מיל רך את s וכן צומת אחד אכו מתקיימות שתי הלמוות.

צע:

1. dequeue: כעת התוור נראה כך $(v_2, \dots, v_r) = Q$, אכו מתקיימים כי או השווינו כפרט יכול להתחילה $c_{[v_2]}$.

$$d[v_r] \leq_{(*)} d[v_1] + 1 \leq_{(**)} d[v_2] + 1$$

- בasher (*) מהנתה האינדוקציה ו(**) מא!
2. enqueue: כעת התו v_0 ורואה $(v_1, \dots, v_r, v_{r+1}) = Q$. נספנו v_0 את הצלמת שיעזא מהתור ובגינו נכנס v_{r+1} . מכיוון $d[v_0] + 1 = d[v_{r+1}]$, נפעיל למקירוט:
- א. אם v_1 היה בטור בעט s_0 יצא אליו בהכרח לפחות $d[v_1] \leq d[v_0]$ ומכיוון נקל כי $d[v_{r+1}] = d[v_0] + 1 \leq d[v_1] + 1$ אז גם $d[v_0] + 1 \leq d[v_1] + 1$.
 - אם v_r היה בטור כשל s_0 יצא אליו לפחות $d[v_r] \leq d[v_{r+1}] \leq d[v_0] + 1$ ולכן גם $d[v_0] + 1 \leq d[v_1] + 1$.
 - אם v_r לא היה בטור כשל s_0 יצא אליו לפחות $d[v_r] = d[v_{r+1}] + 1$ ולכן גם $d[v_0] + 1 \leq d[v_1] + 1$.
1. מתקיימים גם כאלו.
- כ. אם v_1 לא היה בטור בעט s_0 יצא אליו כל הצלמות נכנסו בגיו v_0 והערך $d[v_i]$ שלהס הוא $d[v_0] + 1$ ולכך גם 1 וגם 2 מתקיימים.
- כודרשות.

מסקנה 5: אם $u \in V$ ויצא מ- Q לפחות $v \in V$ כך כי ריצת $BFS(G, s)$ יצא $d[u] \leq d[v]$ (כלומר, ערכי d של הקודקודים יכולים רק לעלות לאורן ההרצה).

הוכחה: נבע ישירות מлемה 4.

лемה 6 (הוכחת נכונות BFS): לאחר הריצת אלגוריתם BFS על גוף מכון $G = (V, E)$ וקוזח $s \in V$, מתקיים:

$$\forall u \in V : d[u] = \delta(s, u)$$

הוכחה: נניח בsvilleה כי קיים צומת אחד לפחות עכשו או שווין. נסמן u הצלמת עם $\delta(u)$ הכי קטן עכשו זה מתקיימים. ככלומר $d[u] \neq \delta(s, u)$. עכשו v הצלמת עם $\delta(v)$ מתקיימת $d[v] = \delta(s, v)$ (ב証明 $s \neq u$ כי $d[s] = 0$ לפי האלגוריתם ואכן $0 = \delta(s, s)$). כאשר v יצא מהתור, הוא עופר על כל שכניו. אם u היה לבן, הוא היה מקבל את הערך הנכון $d[u] \leq d[v] = \delta(s, v) + 1$, וכך u איינו לבן ככלומר v קוזס לו בתור. ולפי מסקנה 5: $d[u] = \delta(s, u)$, כלומר $\delta(s, v) < \delta(s, u)$.

4.3 אלגוריתם סריקת DFS

בහינתן גוף, סריקת DFS על הגרף היא סריקת עולם. הסריקה עוברת על כל הקודקודים של הגרף באופן הבא: כל עוד יש קודקוד שלא ביקרנו בו נברך בו. כאשר מבקרים בקודקוד, בודקים אם יש משchnio טררים ביקרו בו - ואם כן מבקרים בו בקריאת רקורסיבית.

בhaiinten גוף מכון $G = (V, E)$ האלגוריתם DFS סורק את כל הקודקודים. בדומה ל- BFS , האלגוריתם משיך לכל קודקוד צבע שמשמל את מצב הקודקוד:

ה- שחור, ביקרנו סיניינו לטפל בקודקוד.

ש- לבן, טרם ביקרנו בקודקוד.

ו- אפור, ביקרנו אך טרם סיימנו לטפל בקודקוד.

בנוסף לכל קודקוד $V \in V$ ורץ האלגוריתם שלושה ערכים:

א. $d(u)$ - זמן הגעה (צביעה באפור)

ב. $f(u)$ - זמן עזיבה (צביעה בשחור)

ג. (u) פ- קודקוד קודם. השדה פ מגדר לכל קודקוד קודם, אשר אם נסתכל עליו כ"אבא" של הקודקוד הראשון נקבל אוסף של עצים, המכונה גם יער העומק.

תלי בדיקה ספציפית של האלגוריתם DFS ולגרף נתון יכולים להיות מס' יער עומק שונים.

להלן האלגוריתם:

<i>(u)DFS-Visit</i>	<i>DFS(G)</i>
$color[u] \leftarrow g .1$	$u \in V$ for .1
$d[u] \leftarrow t \leftarrow t + 1 .2$	
$v \in adj(u)$ for .3	
$color[v] = w$ if (א)	$color[u] \leftarrow w, \pi[u] \leftarrow \text{NULL}$ (א)
(v)DFS-Visit then (ב)	$t \leftarrow 0 .2$
$\pi[v] \leftarrow u$ i.	$u \in V$ for .3
$color[u] \leftarrow b .4$	$color[u] = w$ if (א)
$f[u] \leftarrow t \leftarrow t + 1 .5$	DFS-Visit(u) then (ב)

האלגוריתם ישמור כמשתנה גלובלי את t : שיאותחל בהתחלה לאפס ויגדל בהתאם לאלגוריתם. t ישמש לציין את זמן הכניסה הנוכחי. האלגוריתם הוא אלגוריתם רקורסיבי שעובר לעומק על כל הגרף.

הגדה: יער העומק מוחזר אליו כמערך π . נשים לב כי *null* במערך מציין שורש של עץ. יער העומק מכיל עצים (מסלולים ארוכים) שונים.

סיבוכיות זמן ריצה: נראה כי כל צומת נקבעת עם $DPS - Visit(u)$ פעמי אחד בדיק. בתחילת, האתחול עולה $O(|V|)$ וסה"כ מבצעים לולאה של $O(|V|)$ כפול $1 + deg(u)$ וסה"כ מקבל $\sum_{v \in V} (1 + deg(v)) = O(|V| + |E|)$ זמן הריצה של האלגוריתם. זמן הריצה לינארי.

הערה חשובה: לא מ투אר כיצד ניתן רצים על האלגוריתם, וכך כל סדר של הקודקודים (כל עוד אנחנו ידעים אותם מראש) הינו חוקי. יתכו ! $|V|$ פרמטריות אפשריות לריצת DFS .

משפט הסוגרים: בכל $G = (V, E)$ על גראף DFS מכוון או שלא מכוון, עבור כל זוג קודקודים $u, v \in V$ מתקיים אחד מהשלושה הבאים (מעבר האינטראול של זמן ההתחלה עד זמן הסיום):

- .1 $[d[u], f[u]] \cap [d[v], f[v]] = \emptyset$
- .2 $[d[u], f[u]] \subseteq [d[v], f[v]]$
- .3 $[d[u], f[u]] \supseteq [d[v], f[v]]$

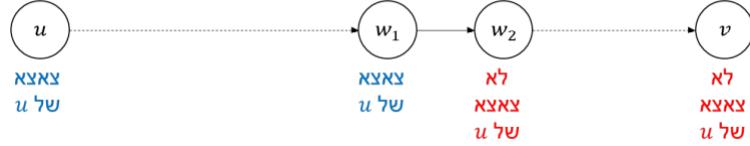
הערה. אם $u \neq v$ בסעיפים 2 ו-3 מדובר על הכהה ממש (לא שווין)

מסקנה: קודקוד v הוא צאצא של קודקוד u ביער העומק אם ומ"מ מקרה 3 מתקיים.

משפט המסלול הלבן: $V \in u$ הוא צאצא של $V \in u$ ביער העומק אם ומ"מ בזמן $1 - d[u] - d[v]$ קיים ב المسلול מ- u לשכל קודקודיו קבועים בלבד.

הוכחה:

אם u צאצא של v ביער העומק, הרו קייס מסלול מ- v לשכל העומק וכפרטו גראף G .
 כל הקודקודים על המסלול הזה הם צאצאים של u , ולכן ע"פ משפט לכל הקודקודים x הללו מתקיים $d[x] < d[u], d[x] \leq d[v]$, כלומר x הפלול הוא לבו וויתנו לראות זאת באלגוריתם שיש קרייה של DFS מ- u על הקודקוד v בלבד.
 נניח כי בזמן $1 - d[u] - d[v]$ קיים מסלול מ- v לשכל קודקודיו קבועים בלבד. נניח כשלילה כי u אינו צאצא של v ביער העומק בזמן $1 - d[u] - d[v]$. נסתכל על המסלול הילך מ- v ויהי w הקודקוד הכי קרוב מ- v על המסלול שהוא צאצא של u וב- w הקודקוד העוקב במסלול (שבהכרח אינו צאצא של u וראה



איור 1: המהשחת הוחוכה

באיור, כיוון ש w_1 יצאא של u מתקיים $d[u] \leq f[w_1]$, $f[w_1] \leq f[u]$ ולכן $d[u] \leq f[w_1]$. האלגוריתם בזק את צבעו של w_2 . מכיוון ש w_2 לנו (כי אחרת זה אומר שכירויו בו ממהלך הסיריקה מ- u - והוא יצאא של u , שהוא עוזרו בתז סריקה זו) אז בהכרח נקבע $d[w_2] < f[w_2]$ ופפיליא $d[u] \leq d[w_1] < f[w_1] \leq f[w_2]$. כלומר $f[w_1] \leq f[u]$ בסתיו להנחתנו. מכיוון בהכרח v יצאא של u כיור העומק.

4.3.1 סיווג קשתות

- ניתן לסווג את הקשתות בגרף בהתאם לריצת ה- DFS כך שכל קשת מסובגת לפי אחד מהסוגים הבאים, כך שאין קשת שנמצאת בשתי קבוצות :
1. קשת עז: קשת מהצורה (u, v) עבור $V \in u$ בלבד. יש לבדוק $k - |v|$ קשתות כאלה באשר k הוא מס' העצים.
 2. קשת אחוריה: קשת מ- v לאב קדמוני של v שנקרה לו u . צריך להתקיים $d[u] < f[v] < f[u]$.
 3. קשת קידימה: קשת מ- v לצאצא לא ישר של v , נקרה לו u . צריך להתקיים $d[v] < f[u] < f[v]$.
 4. קשת חוצה: קשת מ- u לקודקוד שאינו צאצא ואינו אב קדמוני של u . זה כל שאר הקודקודים, של צמתים הקשת זרים.

4.4 גראף מכובן חסר מעגלים (DAG)

גראף מכובן נקרא DAG , כיצד נזהה האם גראף G הוא DAG ?
טענה: גראף G הוא $DAG \iff$ בהרצת DFS אין קשתות אחוריות.
הוכחה: ווכיח קונטראפהוזיטיב.
 נניח שיש מעגל ב- G . \iff נסמן u כזאת ראשונה שנקרה עם $DFS - visit$ במעגל. לכן יש מסלול לבן לשאר הצלחות במעגל. \iff משפט המסלול הלבן, צמותי המעגל הם צאצאים של u ביער העומק ושפחות קשת אחת מהם שחוורת ל- u (מהגדרת מעגל).
 נניח שיש קשת אחוריית u ל- u . בשילוב עם קשתות העץ מ- u (שקיים כי v יצאא של u לפיה הגדרת קשת אחוריית) ונקבל כי יש מעגל בגרף המקורי. וכך.

מסקנה: כתה בהינתן הריצת DFS , נוכל לבדוק בקהלות האם יש קשתות אחוריות (נעבור על כל הקשתות), ואם אין, משמעות הדבר שאין מעגלים. ככלומר אלגוריתם לבדיקה האם יש ב- G מעגל בעלות $O(|E| + |V|)$.

4.5 מילון טופולוגי

הגדרה: גראף מכובן ללא מעגלים (גמ"ל) הוא גראף מכובן שלא מכיל מעגלים.

בהתנתק גמ' ל' נרצה סידור מיוחד של הקודוקדים משמאלי לימי שבו כל הקשותות הן בכיוון משמאלי לימי ומימילא כל מסלול מכובן הוא משמאלי לימי.

הגדה: מינו טופולוגי של קודקודי הגרף $G = (V, E)$ שהוא גמ"ל הוא סידור (v_1, v_2, \dots, v_n) של הקודקודים ב- V כך שכל קשת $e \in E$ מתקיים $j < i$.

1. הרץ DFS (המחשב את $f[u]$ לכל $u \in V$)
 2. החזר את קודקוד V בסדר יורד של $f[u]$

זמן ריצה: ישירות DFS , זמן של $O(|E| + |V|)$

лемה 6: בסיום ריצת $\text{Topoloogical-Sort}(G)$ מתקיים $(v_i, v_j) \in E$ לכל קשת $v_i \rightarrow v_j$.

תהי $v_i, v_j \in E$ וונכון $j < i$ אחרי המיוון הטופולוגי. כולם $f[v_i] > f[v_j]$.
 א. אם $d[v_i] < d[v_j]$ אז בזמן -1 יש מסלול לבן (הקשת) (v_i, v_j) בלבד $v_i \rightarrow v_j$. לכן $d[v_i] < d[v_j] < f[v_i] < f[v_j]$ צאצא של v_i כלומר v_i משלב הירקן v_j עפ"ג משפט המסלול הלבן לפיה השוגרים ישன שתי אפרוריות.
 ב. אם $d[v_j] < d[v_i]$ לפי משפט השוגרים ישன שתי אפרוריות.
 1. במקורה זה אacen המשפט תכף ומתקיים הדרוש $d[v_j] < f[v_j] < d[v_i] < f[v_i]$.
 2. במקורה זה עפ"ג נקבל כי v_i צאצא של v_j בירקן העומק. בפרט קיימים מסלול מכובן $v_i \rightarrow v_j$. אבל, יש לנו גם את הקשת (v_i, v_j) ונקבל מעגל (שרשור המסלול והקשת) בסתריה לכך ש G הוא DAG.

טענה: לגרף קיים מילון טופולוגי \iff הגרף הוא גמל.

4.6 רביים קשרים היבר (G^{SCC})

בגרפים לא מכוונים הגדרנו רכיבי השירותים קבוצות מסוימות כך שקיים מסלול בין כל זוג קודוקדים בקבוצה. בגרפים מכונים המסלולים חיבטים להיות מכוונים.

הגדרה: רכיב קשיר הינו בגרף $G = (V, E)$ מקוון שהוא קבוצה מаксימלית $C \subseteq V$ כך שכולם ב- C מותקנים $v \sim u$ וגם $u \sim v$.

הגדה: גרע $G = (V, E)$ יקרה קשור היטב אם ורק אם G מכיל בדיקון רכיב קשור היטב אחד.

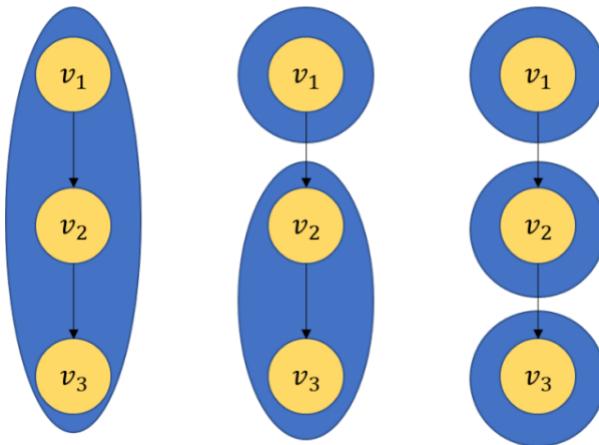
הגדירה: יהי $G = (V, E)$ גרף מקוון. נגדיר את גרף הרכיבים הקשורים היטב כגרף $G^{SCC} = (V^{SCC}, E^{SCC})$ כאשר V^{SCC} היא קבוצת הרכיבים הקשורים היטב של G . בנוסח עבור שני רכיבים $v_1 \in C_1, v_2 \in C_2 \in V^{SCC}$ מותקאים $(C_1, C_2) \in E^{SCC}$ אם ורק אם קיימים קשרים היטב שונים $C_1, C_2 \in V^{SCC}$ מותקאים $(C_1, C_2) \in E^{SCC}$ כך ש- $v_1 \in C_1, v_2 \in C_2$ ו- $(v_1, v_2) \in E$.

הລມה הבהא מבכיעעה על תקונה חשובה של גראף ה-*SCC* שיכולה להיות שימושית במקרים רבים, בעיקר כאשר רצויים להשתמש באלגוריתמים שעובדים רק על גמ"ל (כמו מיוון טופולוגי) עם גرافים פשוטים מעוגלים.

лемה 10: יהי $G = (V, E)$ גרף מכוון (שכמוון יתכוון וככל מעגלים). G^{SCC} הינו גרף מכוון ללא מעגלים. (בג' שיש בו טיגל נקבל שפה ריביב לשירות גודול יותר שיכלנו ליאזר בסיטירה).

נראה אלגוריתם לינארי למציאת רכיבי הקשרות החזקה.

דיון על האינטואיטיבית. נתחל מלהשוו על אלגוריתם DFS . כאשר מרכיבים את אלגוריתם DFS על גראף לא מכון - מקבלים ערך, שבו העצים הם בדיק רכיבי הקשרות של G . (למה? כי מרגע שהפונקציה הראשית הביאה אותנו לקודקוד, אנו סורקים ע"פ משפט המסלול הלבן את כל מי שאיתו באותו רכב. ומצד שני כמובן שלא ניתן להגעה מרכיב אחד לאחר מכן לבסוף לlolאה בפונקציה הראשית). מה יקרה אם נרצה את אלגוריתם DFS על גראף מכון - מרגע שהאלגוריתם מגיע לקודקוד כלשהו, רכיבי קשרות חזקה. באופן דומה למצב בגראף לא מכון - מרגע שהאלגוריתם מגיע לקודקוד כלשהו, הוא יסורך את כל הרכיב החזק שלו (ע"פ משפט המסלול הלבן - כל הקודקודים ברכיב הקשור החזק יהיו בעץ), אולם יתכן שייהיו בעץ גם קודקודים נוספים. ראו דוגמא לחלוקה ע"פ הרצות שונות באירוע. ננסח את הבדיקה באופן פורמלי:



משפט 11: יהי $G = (V, E)$ ויהי $C \subseteq V$ רכיב קשר חזק ב- G . לאחר ריצת DFS על G כל קודודי C נמצאים באותו עץ בירע העומק.

הוכחה: יהי $v \in C$ הקודקוד הראשון שמנינו במהלך DFS מבין כל קודודי C . לכן בזמן $d[v] - 1$ קיים מסלול בין מ"ט לכל קודקוד אחר $u \in C$ ברכיב, וממילא ע"פ משפט המסלול הלבן u יצאא של v בירע העומק, ובפרט נמצא באותו עץ, וממילא כל קודודי C נמצאים באותו העץ.

הערה: נשים לב שככל רכיב היטב יהיה בעץ הירע העומק, אם כן יתכן שכמה רכיבים קשורים היטב יהיו יחד באותו עץ בירע העומק, ועל כך נctrיך להתגבר באלגוריתם. בcut, המטריה היא למצוא שיטה להרייז את DFS בסדר מיוחד שבו נקבל גם את הצד השני - לא רק שככל רכיב קשר חזק נמצא בעץ אחד אלא גם שבלכל עץ יש בדיק רכיב קשר חזק אחד. לצורך כך נגידר מהו גראף משוחך.

הגדרה: יהי $G = (V, E)$ גראף מכון. הגראף המשוחך בגרף G ונסמן $G^T = (V, E^T)$ הוא הגראף המתקבל מהיפוך כל קשת בגרף G . כלומר, $(u, v) \in E^T \iff (v, u) \in E$. באופן שקול, הגראף הוא הגראף שמטריצת השכניםות שלו היא שיחולף של מטריצת השכניםות של הגראף המקורי. נעיר כי האלגוריתם שנראה מוחזיר את קודודי G^{SCC} .

:Strongly Connected Component(G)

א. הרץ $DFS(G)$ המחשב תוך כדי ריצתו את $f[u]$ לכל $u \in V$

- ב. חשב את G^T
 ג. הרץ $DFS(G^T)$ כאשר בולאה הראשית בפונקציה DFS עובר על הקודקודים ע"פ סדר יורד של $f[u]$
 ד. דוח על כל עץ בעיר העומק כרכיב קשור היבט.

מדוע אינטואטיבית הרעיון עובד? ראיינו כבר כי אם נעבור בכיוון מסוימים, למשל בגרף המכוון $c \rightarrow b \rightarrow a \rightarrow c$ מימין לשמאל נקבל אכן שלושה רכיבי קשרות שונים. אם נעבור משמאלו לימין בסיס dfs נקבל רכיב קשרות אחד. נרצה תמיד לעבור בכיוון השני. אנחנו מגדירים את G^T ואז מרצים dfs עלייו לפיה סדר $f[u]$ בסדר יורד. כמובן - זמן הסיום האחרון יהיה ממנו ב- $DFS(G^T)$. נשים לב שטום אלגוריתם אחר לא יעבוד - באינטואיציה נראה כי בהכרח בהינתן שני רכיבי קשרות זמן סיום של צד אחד יהיה לפני השני, ולכן אם נהפוך את הקשת בהכרח באלגוריתם זה לא נשתמש באותו עץ בדומה רכיבי קשרות.

זמן ריצה: DFS עלותו $O(|E| + |V|)$, חישוב G^T עלותו $O(|E|)$ והרצה נוספת של DFS עלותה $O(|E| + |V|)$ וכן ד' עליה עוד זמן לינארי וסה"כ סיבוכיות האלגוריתם $O(|E| + |V|)$.

4.7 מציאת גראף דו צדדי

קלט: גראף דו צדדי $G = (V, E)$ קשור.

פלט: חלוקה של V ל- R ו- L כך שאין קשתות בתוך L ולא בתוך R .

האלגוריתם:

א. הרץ BFS מקודקוד s שרירוטי.

ב. כל מי שבمرחק זוגי היה L וככל מי שבמרחק אי זוגי היה R .

הערה. אפשר כמובן להכליל את הבעיה לגרף לא קשור, ולהריץ את האלגוריתם על רכיבי הקשרות שלו.

טענה (הוכחת נכונות): יהיו L, s . אז לכל $V \in R$ מתקיים $u \in R \iff u, s \in L$.

הוכחה: באינדוקציה על המרחק.

בסיס: עבור $0 = \delta(u, s) = u$ בהכרח $s = u$ ואכן זוגי ולא יכול להיות שהוא ב- R , אם $\delta(u, s) = 1$ אז בהכרח כי לא קיימת קשת ביןיהם, בהכרח $L \in u$ כי הקשת בצד השני.

צעד: נניח שהਮתקיים עבור k . נוכיח עבור $k+1$.

נוכיח עבור k זוגי אחרת הוכחה דומה. יהיו $V \in R$ כך ש- $\delta(s, V) = k$. נסתכל על הקודקוד v הבודם ל- s במסלול הקצר ביותר, נסמןו u . בהכרח $\delta(s, v) = k-1$ ומהנחה האינדוקציה אכן $v \in R$ ולכן $V \in L$ כי הקשת בניהם בהכרח עוברת אל L , qed.

4.8 מציאת עץ מסלולים קצרים ביותר בגמ"ל

קלט: גראף $G = (V, E)$ גמ"ל עם פונקציית משקלים $w : E \rightarrow \mathbb{R}$ וקודקוד מטרה s .

פלט: $SSSP$ (מערך מרחוקים)

נפתחו בתכנות דינמי.

$$f(u) = \begin{cases} 0 & u = s \\ \min_{\{v|(v,u) \in E\}} \{f(v) + w(v, u)\} & o.w \end{cases}$$

נכונות הנוסחה: יהיו $P = (s, v_1, \dots, v_k, u) =$ מסלול קצר ביותר מס u . נשים לב כי הקודקוד v_k שהוא הקודם ל- u מקיים $(u, v_k) \in E$ ולכן נshall כאחת האפשרויות בביטוי שהפונקציה מנשה להביא

למיינום. נשים לב כי העלות של המסלול P היא $w(P) = \sum_{e \in P} w(e) = w(P') + w(v_k, u)$ כאשר $(s, v_k, \dots, u) = P'$ והוא מק"ב בעצמו.
נשים לב כי ביחידה הבאה: 4.9 נציג רעיון די דומה עבור גраф כללי אך זה לא יעזור מהסיבה שמתוארת מטה. מדוע אכן יעבדו? כי אין מעגלים.

עלות זמן ריצה: כדי לחשב את הנוסחה בעילות לכל קודקוד נרצה לשים לב כי ערך הפונקציה של קודקוד תלויה אך ורק בערכי הפונקציה של השכנים הנקנים של הקודקוד. נזכיר שעבור גמל ניתן לחשב מין טופולוגי של הקודקודים שmbטיח לנו שכל השכנים הנקנים של הקודקוד יופיעו בסדר המינו לפניו הקודקוד, ולכן, נוכל לחשב מין טופולוגי ולאתחלה את הערך של הקודקוד s להיות 0. ואז לחשב עבור כל קודקוד את ערך הנוסחה הרקורסיבית עבורו על סדר המינו הטופולוגי. סדר מינו זה מבטיח לנו כי ערכי הפונקציה עבור כל השכנים הנקנים של הקודקוד חושבו לפניו שמנסימים לחשב את ערך הפונקציה עבור הקודקוד ולכן עלות החישוב של הנוסחה עבור כל קודקוד $O(\deg(v))$ במקורה הגורע, על מנת לחיצת הידים קיבל כי עלות זמן הריצה הכוללת של האלגוריתם יחד עם המון הטופולוגי הינה $O(|E| + |V|)$.

4.9 SSSP בגרפים ממושקלים

קלט: גראף $G = (V, E)$ ופונקציית משקל $w : E \rightarrow \mathbb{R}$, $s \in V$.
פלט: $\delta(s, v) \quad \forall v \in V$

4.9.1 נסיוון ראשון - תכונות דינמי

זכור כי במקרה הממושקל:

$$\delta(u, v) := \begin{cases} \min_{p=u \rightsquigarrow v} \{w(p)\} & \exists p = u \rightsquigarrow v \\ \infty & o.w \end{cases}$$

ראיינו בлемה 1, שתת מסלול של מסלול קצר ביותר הוא גם כן מסלול קצר ביותר. זה נכון גם עבור גרפים ממושקלים. אם כן, נשים לב שנוכל לקבל כאן לאלגוריתם רקורסיבי: אם נרצה לחשב את המסלול הקצר ביותר מ- s אל u , ידוע כי שכןו של u הימים u_1, \dots, u_n איזי נחשב את המסלול $s \rightarrow u_1 \rightarrow \dots \rightarrow u_n$ למשיל, ונוסף משקל קשת. כלומר -

$$f(u) = \min\{f(u_1) + w(u_1, u), f(u_2) + w(u_2, u), f(u_3) + w(u_3, u), f(u_4) + w(u_4, u)\}$$

נשים לב כי אם יש מסלול אל u , בפרט ישנו קצר ביותר, והוא בוודאות יעבור קודם لكن אצל אחד השכנים של u (לפי אותה למה). ובמילים אחרות: המסלול הקצר ביותר אל u חייב לעبور אצל אחד מהקודקודים השכנים של u .

נתבונן בנוסחה הבאה, באשר $f(u) = \delta(s, u)$:

$$f(u) = \begin{cases} 0 & u = s \\ \min\{\min_{(v,u) \in E} (\{f(v) + w(v, u)\}), \infty\} & o.w \end{cases}$$

הנוסחה די ברורה, נשים לב לשני דברים:

- א. אנחנו מבערים השוואת בין הביטוי לבין אונסורי - יתכן שלא קיים מסלול בין s ל- u .
- ב. נשים לב כי הגרף מכoon, אנחנו עוברים על הקשותות (u, v) כולם הקשותות שנכנסות אל u .

הלוואי וזה היה קל מאוד. באלגוריתם הנ"ל יש בעיה. מה הבעיה?

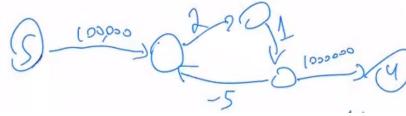
נניח שיש מעגל saw וישנה קשת $s \rightarrow s$. כשנחשב ברקורסיה את המסלול הקצר ביותר מ- s , נצטרך לחשב את (w) כי יש בנים קשת, כשנחשב את (w) נגלה שאנו צורכים לחשב את (u) כי ישנה קשת $w \rightarrow u$, ואז שוב צריך לחשב את (u) . וכשנצטריך לחשב אותו - חזר חלילה. נתקע בולופ: הרקורסיה תקרוס, והובוס שלך יפטר אותה. באסה.

מסקנה: בגרף מכוכו, ללא מעגלים, נסחה זו תעבור בסיבוכיות $O(|V| + |E|)$.

4.9.2 סוגי מעגלים

כפי שראינו, מעגלים עושים לנו בעיות. ישנו מספר סוגים של מעגלים:

a. מעגל שלילי - מעגל כמו המעגל המתואר כאן עליל והוא מעגל שלילי. הליכה על מעגל שכזה תמיד תהיה טובה לנו כי נعود $+1$ ויריד -5 - קלומר בכל סיבוב אנחנו נהוו -2 . במצב זה, אם נלך אונסוי פעמים על המעגל נגיע למשקל שהוא מאוד נמוך - מינוס אונסוי.



לכן, אם יש מעגל שלילי מס l ב- G נגיד: $\infty = (u, s, \delta)$.

בעת, נניח כי בגרף יכולים להיות משקלים שליליים אך אין מעגלים שליליים.

b. מעגל אפס - מעגל שסכום המשקלים שלו הוא אפס, ואין סיבה לעובר בו. למשל מעגל כמו שמתואר מעלה, במקום 1 יהיה הערך 3. נראה כי סכומו יהיה $0 = 5 - 5 + 3 - 2$. מעגל כזה לא מפיע לנו.

g. מעגל חיובי - מעגל שסכום הערכים על הקשתות שלו חיובי. אין סיבה לעبور עליו במקרה מסויל קוצר ביותר.

מסקנה: אם אין מעגלים שליליים, ניתן להניח שהקיים מסלול קצר ביותר שהוא מסלול פשוט. (מדובר? כי לא נרצה לעبور במעגל חיובי, ועל מעגל אפס אפשר לדלג). מכאן, ניתן להניח כי במסלול קצר יותר יש לכל היותר $1 - |V|$ קשתות. (זהו מסלול פשוט: אין בו מעגלים, ולכן ניתן לעبور לכל היותר $|V| - 1$ פעמיים). במקרה הגורע ביותר עברנו על כל $|V|$ הקודקודים, יש בנים $1 - |V|$ קשתות).

נשים לב - גם אם אין מעגלים שליליים, אנחנו עדין באוטה בעיה שנתקלנו בה באשר ניסינו לעבור בתוכנות דינמי. אנחנו לא מפרקים את הבעיה לתתי בעיות שם - תמיד נתקע בולופ.

4.9.3 הקלת קשותות – Relaxations

רעיון האלגוריתם: לכל $V \in u$ האלגוריתם מתחזק ערך $d[u]$ שהוא חסם עליון על (u, δ) . קלומר, $d[u] \geq \delta(u, u)$: האלגי עלי פניו ריצתו יוריד ערכי d עד ש(בתקווה) כל ערכי d מקיימים:

$$\forall u \in V : d[u] = \delta(s, u)$$

כמו כן, נשימוש במשתנה $[u]$ להגדיר את עצ המסלולים הקצרים ביותר.
כך יראה האתחול:

INITIALIZE-SINGLE-SOURCE($G = (V, E), s$)

- 1 **for** each vertex $v \in V$
- 2 $d[v] \leftarrow \infty$
- 3 $\pi[v] \leftarrow \text{NULL}$
- 4 $d[s] \leftarrow 0$

** נשים לב כי $d[s] = 0$ היא הנחה לגיטימית, לעומת $\delta(s, s) = 0$, הדריך היחידה ש $\delta(s, s) \neq 0$.
היא שיהיה מעגל שלילי שמתחל וונגמר בס, ואז < 0 , אך אנחנו מניחים שאין מעגלים שליליים.

נשים לב: אנחנו עובדים לפי קונבנצייה ש $d[s]$ מצביע על *null* בפועל, וכן יתכנו קודקודים נוספים שאינם מצביעים על *null* (מי שלא נניחים אל s), אך נבדיל בהםים לבין s ה $d[s]$ שלהם יהיה אינסופי ו-0. $d[s] = 0$ ו- ∞ .

למה אי שוויון המשולש (במקרה הממושקל): עבור $G = (V, E)$ עם פונקציית משקל $w : E \rightarrow \mathbb{R}$ וכן $s \in V$ ו- $u, v \in V$ מותקיים:

$$\delta(s, v) \leq \delta(s, u) + w(u, v)$$

מסקנה: נניח כי $\delta(s, u) \geq \delta(s, v)$

$$d[u] + w(u, v) \geq \delta(s, u) + w(u, v) \geq \delta(s, v)$$

אי, אם $d[v] > d[u] + w(u, v)$ אז ניתן להקטין את $d[v]$ להיות $d[u] + w(u, v)$. מדוע? כי במקרה זה, כיוון ש $d[u] + w(u, v) \geq \delta(s, v)$, ודין אם נגדיר $d[v] = d[u] + w(u, v)$ יתקיים $d[v] \geq \delta(s, v)$ כפי שרצינו. פועלות הקטנה זו נקראת **פעולת הקלה**.

(עוד אינטואיציה: נניח כי $d[u] = 100, d[v] = 200$, וכן ישנה קשת $e : u \rightarrow v$ ש- $w(e) = 30$.
אכן מותקיים $d[v] > d[u] + w(u, v)$, ולכן כדאי לשפר את $d[v]$ ולהקטין אותו להיות המסלול של $d[u]$ ועוד הקשת ששויה 30, שכן ערך המסלול ירד.)

פסאודו לפעולות relax - הקלה:

RELAX($(u, v), w$)

- 1 **if** $d[v] > d[u] + w(u, v)$
- 2 $d[v] \leftarrow d[u] + w(u, v)$
- 3 $\pi[v] \leftarrow u$

נשים לב כי אם בחרנו לבצע הקלה, שינו את אבא של v להיות u ולכון π משתנה.

4.9.4 אלגוריתם מבוסס הקלות והוכחת נכונות של בלמן פורד

אלגוריתם מבוסס הקלות הוא אלגוריתם שמאתחל ערכיו d, π עם אוחת בערת קריה ל-*Initialize*, ולאחר מכן כל העדכוןים לערכיו d יבצעו רק בערת פעולות של *Single – Source relax*

המטרה: להראות כי האלגוריתם מבצע סדרה של הקלות שימושיות לכך ש:

$$\forall u \in V : d[u] = \delta(s, u)$$

כל הטענות הבאות יהיו נכונות לכל אלגוריתם מבוסס הקלות.

лемה 7 (лемת חוסר המסלול): לאחר ביצוע $ISS(G, s)$ (אתחול) באלגוריתם מבוסס הקלות, אם אין מסלול מ- s ל- $v \in V$ אז תמיד מתקיים $\delta(s, v) = d[v] = \infty$.
הוכחה: הרכבה נקבע ושוות למספר 9 שכך לא מטה. געת האתחול ותכצע $\infty = [v]$. כמו כן, אם אין מסלול מתקיים $\infty = \delta(s, v) = d[v]$ ולפי lemma 9 מהרגע זהה מתקיים הערך לא ישנה יותר, וכן מתקיים כדרש.

лемה 8 (лемת הקלת מסלול): נניח כי המסלול $P = (v_0, \dots, v_k)$ הוא מסלול קצר ביותר עם $v_0 = s$. נניח שבזמן ריצת אלגוריתם מבוסס הקלות לאחר ביצוע $ISS(G, s)$, סדרת הקלות שהאלגוריתם מבצע מכילה את סדרת הקשות של P כתת סדרה לפי סדרן P . (כלומר, לאחר ביצוע פעולות רילקס על כל (v_i, v_{i+1}) ($i \leq k - 1$) לפि השדר **אך לא בהכרח ברצף** איזו, בסיסם ריצת האלגוריתם מתקיים $d[v_k] = \delta(s, v_k)$ (כלומר, הבעה נפתרה עבור קודקוד v_k).
הוכחה: וראה כי לאחר ביצוע הקללה על הקשת ה- i (כתת הסורה) מתקיים $d[v_i] = \delta(s, v_i)$. ווכיה באינדוקציה על i .

בבסיס: $i = 0$, ככלומר משפטות הדבר היא $s = v_0$, אכן בתחלת מתקיים $d[s] = \delta(s, s) = 0$. ע"פ lemma 9, הערך של $[s]$ לא השתנה עוד לעומת $d[s]$ לא השתנה עוד לעומת $d[v_k]$.
 צעד: ע"פ הטענה האינדוקטיבית, לאחר הקללה על הקשת ה- i בתת סורה מתקיים שכיצענו הקלות על הקשת הראשונית, השווייה וכו' עד ה- i וכן מתקיים $d[v_k] = \delta(s, v_k)$.
 בקללה ה- $i + 1$, לפי lemma 10, מתקיים כי $d[v_{i+1}] = \delta(s, v_{i+1})$, כי אכן ביצעו הקללה על כל הקשות הבודדות ובפרט על זו ה- i , וכן לאחר ביצוע הקללה יתקיים השווון. כנדרש.

נשים לב. lemma 8 טעונה שאם יש לי את המסלול הקצר ביותר, איזו איזי יווע שפטורי את הטעיה עכו v_k . מדע צריך את זה? נסתכל על המשפטה הבא –
מסקנה: נשים לב כי lemma 8 ולפי lemma 1, אם פרטנו את הבעה עboro v_k , איזי פרטנו את הבעה לכל $0 \leq i < k$, ככלומר $d[v_i] = \delta(s, v_i)$ $\forall i < k$. ככלומר, אם נרץ פעם אחת נפטרו את הבעה עboro v_1 . אם נרץ פעמיים, נפטרו את הבעה עboro v_2 . וכן אם נרץ עboro $-V$ נקבל את הפטרון להבעה.

лемה 9 (лемת החסם העליון של $d[v]$): באלגוריתם מבוסס הקלות, לאחר ביצוע האתחול $ISS(G, s)$ לכל קודקוד $V \in v$ תמיד מתקיים $d[v] \geq \delta(s, v)$. בנוספ', מהרגע שבו האלגוריתם מציב $d[v] = \delta(s, v)$, הערך של $d[v]$ לא ישנה יותר עד לסיום ריצת האלגוריתם.

הוכחה: ווכיה באינדוקציה על מס' פעולות הקללה שהאלגוריתם מבצע. נספחים n .
 בסיס: $n = 0$. $\forall v \in V/s$ אם האלגוריות לא ביע עדיין הקלות מתקיים $d[v] = \infty \geq \delta(s, v)$.
 וко עכו $s = v$ מתקיים $d[s] = 0 = d(s, s)$ לפי הגדרת האתחול, ומתקיים $\delta(s, s) = 0$ כי איזו מעגליים שליליים.
 צעד: נניח שלפני ביצוע הקללה $d[x] \geq \delta(s, x)$ $x \in V$ מתקיים לכל $(u, v), w$ מתקיים $Relax((u, v), w)$.

אם פעולות ה- *relax* לא שימתה שום ערך של d , לפי הנחת האינזוקציה עדין מתקיים כי : $\forall x \in V$

$$d[x] \geq \delta(s, x)$$

אם פעולות ה- *relax* כן שימתה ערך של d , היא יכולה לשנות רק את הערך של $d[v]$, כפרט ותקיוןicut כי $(v, u) = d[u] + w(u, v) \geq \delta(s, u)$, מתקיים כי לפי הנחת האינזוקציה כי $(u, v) = d[u] + w(u, v) \geq \delta(s, v)$, ולכן נקבע $d[v] = d[u] + w(u, v) \geq \delta(s, u) + w(u, v) \geq \delta(s, v)$ כאשר (*) מתקיים לפי או שווין המשולש. סה"כ $d[v] \geq \delta(s, v)$ כנדרש.

נשים לב כי עליינו להוכיח דבר מסוים, מהרגע שיציב ערך $d[v] = \delta(s, v)$ הערך של d לא ישנה יותר. אס"כ, כיוון שפעולות *relax* ורק מתקיימות ערכי d , והולemo הרוי כי $d[u] \geq \delta(s, u)$ מתקיים $d[u] = \delta(s, u)$ לא יכול להשתנות יותר כי $d[u] \geq \delta(s, u)$ לא יכול לגוזג, $d[u] \geq \delta(s, u)$ להיות קטן יותר כי $d[u] \geq \delta(s, u)$, סה"כ ערכו לא משתנה.

лемה 10 (תבונת ההתכנסות): נניח שהמסלול $v \rightarrow u \sim s$ הוא מסלול קצר ביותר מס v . אז, לאחר ביצוע $ISS(G, s)$ באלגוריתם מבוססת הקלות, מתקיים שאם $d[u] = \delta(s, u)$ מתקיים $d[v] = \delta(s, v)$ לפני ביצוע הקללה על הקשת (u, v) , אז לאחר ביצוע הקללה על הקשת מתקיים $d[v] = \delta(s, v)$ (כלומר, אם המסלול מס $\sim s$ כבר ידוע לנו, לאחר ביצוע הקללה אחת אנחנו נדע את $v \sim s$ כי הרוי סה"כ מושגים הличה על קשת אחת).

הוכחה: ע"פ למה 9, תמיון מתקיים $d[v] \geq \delta(s, v)$, לפחות למקורות. אס"כ לפיו ביצוע הקללה התקיים $d[v] = \delta(s, v)$ אז סימנו כי הערך לא משתנה יותר לפי لما .9

ב. אחרת, ככלומר $d[v] > \delta(s, v) = \delta(s, u) + w(u, v) = d[u] + w(u, v) > \delta(s, u)$. מכיוון $d[v] > \delta(s, v)$, זה בזיהוג התנאי $d[v] > d[u] + w(u, v)$ מתקיים $d[u] + w(u, v) = \delta(s, v_1) = d[v_1] = \delta(s, v_1)$, וכך אמינו מגדכים לאחר ביצוע הקללה את v (בוגרף, $d[v_1] = d[u] + w(u, v) \geq d[u] = \delta(s, u)$ וכן $d[u] = \delta(s, u)$)

$$d[v] = d[u] + w(u, v) = \delta(s, u) + w(u, v) = \delta(s, v)$$

ומלמה 9, ערך $d[v]$ לא משתנה עוד. כנדרש.

4.9.5 האלגוריתם של בלמן פורד

נשים לב כי לפי למה 8, אם נדע את המסלול הקצר ביותר מס אל v נוכל לדעת את (v, s) . ובכך, לשם כך נצורך לפתור את הבעיה עצמה - המסלול הקצר ביותר. עם זאת, נשים לב כי אם נבצע הקללה לכל קשותות הנגרף - לכל קשת פעם אחת: בזודאות ביצענו הקללה על הקשת הראשונה במסלול הקצר ביותר, אך לא ידוע לנו באיזה שלב של הקלקות ביצענו עליה הקללה. ככלומר שאנו ח纠正 יודעים כי הקללה בקשת $v_2 \rightarrow v_1 \rightarrow v$ ולפי אותה لما נדע כי $d[v_2] = \delta(s, v_2) = d[v_1]$. באופן דומה ומחרוזי: נבצע את התהילה שוב ושוב, עד שנקבל לאחר k איטרציות - עברו כל קודקוד s שיש אליו מסלול קצר ביותר מה $d[u] = \delta(s, u)$ עם לכל היתר k קשותות, יתקיים $d[u] = \delta(s, u)$.

כמה איטרציות אנחנו צריכים? בהינתן ההנחה שאון מעגלים שליליים, לכל קודקוד $V \in u$ קיים מסלול קצר ביותר מס אל u שהוא מסלול פשוט, במסלול כאח יתקיים כי אורך $\geq |V| - 1$. ולכן נצורך לבצע לכל היתר $|V| - 1$ איטרציות.

הערה. אנחנו תמיד נדע את $d[v_1]$ בהתחלה כי יתקיים התנאי של אי השווון לפי האלגוריתם כי $d[s] = 0$, זה לא בהכרח יקרה עבור שר הקודקודים כי נקבע אי שווון של $\infty < () + \infty$ שאינו נכון בהרמתה.

להלן האלגוריתם:

בתחילת, נאתחל. לאחר מכן במשך $|V| - 1$ פעמים אנחנו נעבור על כל הקשותות בוגרף ונבצע הקללה.

BELLMAN-FORD($G = (V, E), w, s$)

- 1 INITIALIZE-SINGLE-SOURCE($G = (V, E), s$)
- 2 **for** $i = 1$ to $|V| - 1$
- 3 **for** $(u, v) \in E$
- 4 RELAX($(u, v), w$)

סיבוכיות זמן הריצה: $O(|V| \times |E|)$

בנייה האלגוריתם: נניח וקיים מסלול, אז קיים גם מסלול קצר יותר (אחרת ראיינו כי הערך יהיה אונסוי' כבר בהתחלה, ולא ישתנה), אנחנו יודעים לחשב את $d[v_1]$ במסלול לפי הערה כאן לעיל, ומשם לפחות למה 10, לאחר ביצוע הקלה אחת אנחנו יודעים גם את $d(s, v_2) = \delta(s, v_2)$. וכן הלאה ממשיכים עם הקלה ואח"כ לפחות 10 ולבסוף אנו יודעים את $d[v]$.

נשים לב: ב-DAG ישנו מין טופולוגי, ולכן ניתן להחליט שנעבור בסדר מסוים על הגראף. וכך, לא נctrיך בכל שלב באלגוריתם בשורה 3 לעבור על כל הקשתות, וכל עבור בכל קודקוד רק על הקודקודים שוויצאים ימינה. נשים לב כי לקודקוד הראשון יש אפוקטיות לכל רק' ימינה וכן הלאה על החבאים. נקבל כי סיבוכיות זמן הריצה היא $O(|E| + |V|)$.

הערה סופר חשובה: כיוון שאמרנו שמסלול ארוך ביותר יהיה בגודל $1 - |V|$ כי הוא פשוט (לא נרצה לעبور על מעגלים). ניתן למצוא דרך לאלהות מעגל שלילי – ניתן להרצץ במלן פורד ולהוסיף איטרציה מס' $|V|$. לעבור בה על כל הקשתות. אם אכן אפשר לעשות הקלה לאחת הקשתות יש נחזר שקיים מעגל שלילי. מודוע? כי אם אפשר לשפר, מתוכנה זו בדיק, זה מסלול בגודל $|V|$, מעגל שלילי.

4.10 האלגוריתם של דיקסטרה

4.10.1 הגדרת הבעיה ומבוא

קלט: גראף $G = (V, E)$ ממושקל עם פונקציית משקל $E \rightarrow \mathbb{R}^+$: w (משקלים חיוביים בלבד),
וקודקוד מקור $s \in V$
פלט: $\delta(s, u) : u \in V$ וכן להציג עץ מסלולים קצר ביותר מס'.

באלגוריתם של פרים, למציאת MST, בכל איטרציה האלגוריתם בוחר את הקשת בעלת המשקל הנמוך ביותר שחווצה את החתק ומוסיף אותה לעצ. האלגוריתם של דיקסטרה פועל באופן דומה. בכל איטרציה, האלגוריתם של דיקסטרה בוחר את הקשת הטובה ביותר להוסיף לעץ המסלולים. השוני בין האלגוריתמים – מה זה אומר "טובה ביותר"?

הגדרה: נסמן ב- S את קבוצת הקודקודים שנמצאתרגע בעץ המסלולים הקצרים ביותר. מסלול P יקרא "מסלול מיוחד" אם כל הקודקודים במסלול P נמצאים ב- S חוץ מהקודקוד האחרון שלו נמצא ב- S . ייתכנו כמה מסלולים מיוחדים P כנ"ל.

האלגוריתם של דיקסטרה בכל איטרציה בוחר את הקודקוד u , $S \neq u$ כך שיש מסלול מיוחד מס' l שכרג עז הוא מסלול מיוחד קצר ביותר ביחס לכל המסלולים המקיימים. כמובן, אנחנו בוחרים בוחרים קודקוד u שלא נמצא בתוך קבוצת הקודקודים בעז, והקודקוד u זה שאנו בוחרים הוא זה שהמסלול מס' l

אליו הוא הקצר ביותר ביחס למל המסלולים המוחדים. זהי נקודה מהותית - הוא בוחר ביחס לכל המסלולים המוחדים, לא ביחס לכל מי שמשתאים בס. האלגוריתם מוסיף את הקשת الأخيرة על המסלול (זאת שנראית כרגע "הטובה ביותר") המוחד P לעז ומוסיף את s ל- S .

נחיד: אנחנו נסתכל על קבוצה S שהיא כל הקודוקדים שגרע בעץ המסלולים הקצרים ביותר. נרצה בכל פעם לבחור את הקודוקוד שהוא לא S , והמסלול הקצר ביותר אליו הוא הקצר ביותר וכן המסלול מס' אל אותו קודוקוד הוא מסלול שכל הקודוקודים בו נמצאים S , פרט לקודוקוד האחרון שמנצא $S \setminus V$ והקשת الأخيرة במסלול היא קשת שהווצה את החתק.

4.10.2 האלגוריתם

האלגוריתם של דijkstrra הוא אלגוריתם מבוסס הקלות, אך הוא מוחזק ערכי d וכן מבצע ISS בתחלת הרצה ומוריד את ערכי d רק באמצעות $relax$. נשים לב כי מתחילה את מבנה הנתונים Q עם s ובתחילה s יצא ממנו. המפתח יהיה $[u]$ לכל קודוקוד. להלן האלגוריתם:

```

DIJKSTRA( $G = (V, E)$ ,  $w$ ,  $s$ )
1 INITIALIZE-SINGLE-SOURCE( $G = (V, E)$ ,  $s$ )
2  $S \leftarrow \emptyset$ 
3  $Q.Init(V)$ 
4 while  $Q \neq \emptyset$ 
5    $u \leftarrow Q.extract\_min()$ 
6   Add  $u$  to  $S$ 
7   for each  $v \in ADJ[u]$ 
8     RELAX( $(u, v)$ ,  $w$ )

```

אלגוריתם מבוסס הקלות מתחילה ב(s) ויצירם קבוצה ריקה S . כמו כן, אנו משתמשים בתווך קדיימות Q . מתחילה אותו ראשית. נגידר כי מפתחות התווך הם ערכי d . בכל שלב, כל עוד התווך לא ריק (בתחילה הוא מאותחל עם כל איברי V כל הקודוקודים), אנחנו נוציא את זה עם המפתח המינימלי (ה- d הכי קטן, המרחק הכי קצר...), נכנסו אותו אל S , וונבור על כל שכניו וنبצע להם הקללה.

ישנה נקודה שצורך לשים לב אליה - אם הערך של $d[v]$ יורך, משמעות הדבר היא כי צריך לבצע $decrease_key$ על v ב- Q . כמובן, ישנה את פעולה $relax$ שאם נכנסים אל if בתוך רילקס, אז $.Q.dec_key(v, d[v])$.

הקודוקוד הראשון שיצא מ- Q הוא $s = d[s] = 0$ ושל כל השאר הוא אנסוף. וממנו יבנה העץ.

סיבוכיות זמן הרצה: נראה כי יש לנו אתחול, ידוע כי הוא עולה $O(|V|)$ זמן. הלולאה מתחכעת $\sum_{v \in V} deg(v) = O(|E|)$ פעמים ולכן כמכלול ($deg(u)$ פעמים מס' הפעמים $|V|$) פעמים, חלק 7 מתחכע ($deg(u)$ פעמים ולכן כמכלול ($|E|$ פעמים מס' הפעמים $|V|$)) פעמים.

שיכולה להתבצע שורה 8, וכן בכל אחד מפעולות ה-*relax* אנחנו יכולים לבצע בצע *decrease-key*. מכאן :

$O(V)$	<i>שיעלה ISS</i>
<i>init</i>	
$ V $	<i>פעמים extract-min</i>
<i>decrease-key</i>	<i>פעמים E </i>
זמן הריצה הוא:	

$$O(\text{init} + |V| + |V| \times \text{extract-min} + |E| \times \text{decrease-key})$$

ערימה ביןראית:

$$O(|V| + |V| + |V| \times \log(|V|) + |E| \times \log(|V|)) = O(|V|\log(|V|) + |E|\log(|V|))$$

ערימות פיבונאצ'י:

$$O(|V| + |V| + |V| \times \log(|V|) + |E| \times O(1)) = O(|V|\log(|V|) + |E|)$$

הערה חשובה: האלגוריתם מאותח בהתחלת את Q להיות כל קבוצת הקודקודים. תמיד בתחלת האלגוריתם s יצא קודם!! בכל שלב נוציא את האיבר שהמפתח שלו הכי קטן. לבסוף נסימן לא איברים בתור קדימות ונסיים.

4.10.3 הוכחת נכונות של דיקסטרה

האלגוריתם של דיקסטרה הוא אלגוריתם חמדני. ונitin להוכיח למה בחירה חמדנית וכו'. אך הפעם השתמש בהוכחה יותר ישירה. **הערה.** אינוריאנטה היא טענה שנרצה להוכיח שתמיד מתקינה.

אינוריאנטה 1: כל פעם שהאלגוריתם של דיקסטרה מגע לשורה 4 בפסודו קוד, אז לכל u מתקיים $d[u] = \delta(s, u)$ (כלומר, כל מי שהכנסנו כבר לקובוצה S ונכנס לעז, אנחנו יודעים את המרחק הקצר ביותר עבורי. אם זה נכון, זה בפרט יהיה נכון כאשר מגע לשורה 4 בפעם الأخيرة, ואז יתקיים לכל הקודקודים כי $d[u] = \delta(s, u)$ והוכחנו את הטענה).

הוכחה: ההוכחה היא באינדוקציה על מס' הפעולות שהאלגוריות בראיצה מסוימת מגע לשורה 4. נספנו n .

בסיס: $n = 0$, כשהגשים את כל מה שכוצע עד כה באלגוריתם היה אתחול בלבד. הקבוצה S ריקה וכפיסט האינוריאנטה מתקיימת כמובן ורק **צעד:** נראה שכל פעס שהאלגוריות מוסר קווקז $s \rightarrow u$ מתקיים $d[u] = \delta(s, u)$. יחד עם لما נקבע שמהוגע שבו u התווסף אל S תFINE וקיימים $d[u] = \delta(s, u)$. יהל עס למיה נីיה בשיליה כי u מתווסף אל S גודלה, אך $(u, d[u]) \neq \delta(s, u)$. בהיל u הוא קווקז הראשון שתווסף אל S שעכוו $(s, u) \neq \delta(s, u)$. וכך כל הקוווקזים שקדמו לעז, יקיימו $d[k] = \delta(s, k)$. נשים לב כי $s \neq u$ כי כבר באתחול $0 = \delta(s, s) = d[s]$ (כי אין מעגלים שליליים כי אין קשתות עס משקל שלילי) ולפי لما הערץ לא ישנה לעולס. מכאן שלא ותכו $s = u$.

מכאו נסיק כי לפוי ש התווסף אל S , בזוחאות $\emptyset \neq S$ (בזוחאות s יהיה שס). כמו כן, בהכרח קיים מסלול u אל s כי אחרת לפי ליפה 7 (לט חסר המסלול) ותקיים כי $\delta(s, u) = \infty$, וכן בזוחאות יש מסלול $u \sim s$. ככלור $\infty < \delta(s, u)$.

נסמו P' מסלול קצר ביותר מס אל s ב- G (קיים כזה), נסמו בע את הקוזוקד הראשון ב- P' שנמצא S/V . בהחלט יתכו כי $s = u$. כמו כן, נסמו את הקוזוקד לפני u במסלול x , וכן $s \in x$. ככלור המסלול הווה $(s, x, y, \dots, x, y, \dots, x)$ ($P' =$ מכאו שבעצמו ש הцентр של S , או יודיעים כי $d[x] = \delta(s, x)$ כי הוכחנו ש הוא הראשו עכשו או השווינו קרה).

נסתכל על מה האלגוריתם עשה ברגע x הцентр של S : האלגוריתם שלג ההוא עבר על כל הקשחות שיצאו מ- x וכברט על הקשת (x, y) וקבע לעילו הקלות. ליפה 10 נובע, שלאחר הקלה על הקשת (x, y) נקלל כי $\delta(s, y) = \delta(s, x)$ (P' הרושה של y שפטותית בע y הוא מסלול קצר ביותר מס אל s , וכברט תת מסלול קצר ביותר הוא מסלול קצר ביותר, וכן המסלול עד x הוא הקצר ביותר, ומכוון לפוי למה קוזומת).

מכאו, $d[u] \leq \delta(s, u) \leq \delta(s, y) \leq d[y]$ (d כי אנחנו לא משקלים שלילים, והמסלול יכול רק לנזר) מצד שני, האלגוריתם בחר בס הцентр אל S כאשר y הוא אחת מהאופיות, ולכן בשלב זה $d[u] \leq d[y]$ (d (y הינה נבחר) - שיס לג כי y הגודר להיות $S \setminus V$ וכן בהכרח u יצא קוזס מהתו לפוי y).

סה"כ קובלנו $d[y] = \delta(s, y) \leq d[u] \leq d[y]$ וכן $d[u] = d[y] = \delta(s, u)$ (d כי אוניברסיטי מתקיים כי כל השויות במאצע מתקיימים ונקלל $d[u] = d[y]$ בסתיויה ולכן לנו שווינו ביחס. כאמור.

4.11 סיכום

ישנם שלושה אלגוריתמים שונים לבניית *shorts paths*.
אלגוריתם BFS: מטפל ב-*SSSP* במרקחה הלא ממושקל. סיבוכיות זמן הריצה שלו $O(|E| + |V|)$
האלגוריתם של בלמן פורד: מטפל ב-*SSSP* במרקחה הממושקל. מניח שיתכנו משקלים שלילים אך לא יתכנו מעגלים שלילים. סיבוכיות זמן הריצה שלו $O(|E| \times |V|)$
האלגוריתם של דיקסטרה: מטפל ב-*SSSP* במרקחה הממושקל, אך מניח שלא יתכנו משקלים שלילים. סיבוכיות זמן הריצה שלו $O(|V| \log |V| + |E|)$ עם פיבונacci ועם עירומה ביארית $O(|V| \log(|V| + |E|))$. **אלגוריתם חמדן.**

5 הרצתה 5: shorts path – APSP

5.1 מבוא לכפוף מטריצות מהיר

קלטי: שתי מטריצות $A, B \in \mathbb{F}^{n \times n}$. נסמן: $C = A \times B = \sum_{i,j} c_{ij}$
פלט: לחשב את $C = A \times B$, ככלור את

זרץ נאיית לפרטון הבעה: $C_{ij} = \sum_{k=1}^n A_{ik}B_{kj}$, כל תא יעלה לחישוב לפי הנוסחה $O(n)$ והמטריצה בגודל $n \times n$ ישם n^2 תאים ומכאן עלות האלגוריתם $O(n^3)$.

אתה מהשאלות החשובות בעולם התאוריה של מדעי המחשב, היא מהו הזמן הכى מהיר שבו ניתן לחשב את המטריצה C .
הבחנה ראשונה: זמן מינימלי לפתרון הבעה הינו $\Omega(n^2)$ כיוון שגודל הפלט הינו $O(n^2)$. וכן הוא לינארי בגודל הקלט שלו.

האלגוריתם של שטרסן: רץ בזמן $O(n^{2.81}) = O(n^{\log_2 7})$ - האלגוריתם שביצענו בעצמנו בתרגילים (1) שאליה (1), מגדירים כפליים חדשים ומצלחים להציג לנוסחת הנסיגה $T(n) = 7T(\frac{n}{2}) + O(n^2)$.

בשנת 1986 נפל דבר, ו-*Coppersmith-Winograd* הצליח להציג לסיבוכיות זמן של $O(n^{2.376})$. לאחר מכן הסיבוכיות ירדה ל- $O(n^{2.37287})$

נסמן בא את האקספוננט של האלגוריתם הכי מהיר שקיים לפתרון הבעיה. בהכרח, $\leq 2.37287 \leq \omega$ כי כיום ידוע אלגוריתם שפותר בזמן זה, וכן נקלט סה"כ כי:

$$2 \leq \omega \leq 2.37287$$

מכאן, האלגוריתם הכי מהיר לכפל מטריצות ריבועיות מגודל $n \times n$ עולה (ω זמן).

אנחנו לא נלמד על כפל מטריצות מהיר בקורס. עם זאת, ינסם הרבה אלגוריתמים שימושיים בכפל מטריצות מהיר בתורת פרווציאורה. ככלומר, "קופסה שחורה", אליה נכנס קלט, מתבצע *FMM* (*Fast Matrix Multication*) ויצא פלט. השתמש כנהחה שכפל מטריצות עלותה (ω זמן) $O(n)$ ועזר בכך להוריד זמן ריצה של אלגוריתמים.

עתabraut נראתה את האלגוריתם של סיידל, אשר $G = (V, E)$, מטריצת משקלים w . מינימום כי V מטריצות בואפן שייר.

5.2 הגדרת APSP

קלט: גראף $G = (V, E)$ מכובן/ לא מכובן ופונקציית משקלים w . מינימום כי $V = \{1, \dots, n\}$.

פלט: שתי מטריצות

1. מטריצת מרחוקים $D = (d_{i,j})$ מוגדר $d_{i,j} = \delta(i, j) |V| \times |V|$ כך ש(i, j) מושך,
2. מטריצת קודמים $\pi = (\pi_{i,j})$ מוגדר $\pi_{i,j} = \pi$ מושך $|V| \times |V|$ כך ש(i, j) מושך אם $i = j$ או שאין מסלול מושך בין i, j , אחרת $\pi_{i,j}$ הוא קודקוד שקדם ל- j במסלול הקצר ביותר מ- i .

פתרון נאיבי: אם יש קשותות שליליות אך אין מעגל שלילי, להריץ את בלמן פורד מכל קודקוד ולקיים זמן ריצה של $O(|V| \times |E|) \times |V| = O(|V|^2 |E|)$.

הנחה: נניח כי הנגרף מיוצג ע"י מטריצת שכנוויות.

5.3 האלגוריתם של Floyd – Warshall

הערה: האלגוריתם יניב שיטות שליליות, אך אין מעגל שלילי.
האלגוריתם פותר את הבעיה באמצעות תכונת דינמי. נרצה לחשב $|V|^2$ מרחוקים – בין כל זוג (i, j) . נרצה להגדיר נוסחת נסיגה לכל מסלול שבעל שלב מותבسطת על קלט קטן יותר. הקלט יהיה הקודדים בהם ניתן להעזר במסלול.

הגדרה: יהיו $P = (v_1, \dots, v_\ell) = P$ מסלול. אזי, הקודדים $v_{\ell-1}, v_{\ell-2}, \dots, v_1$ הם קודודי ביניים של P .

נגדיר את נוסחת הנסיגה באמצעות קודדים הביניים בהם ניתן להשתמש. ככלומר, עבור הרמה k אנו רוצים לדעת את אורך המסלולים הקצרים ביותר מבין זוגות המסלולים בגרף כאשר מותר להיעזר בקודודי ביניים רק בקודדים $\{1, \dots, k\}$. נראה כי P מסלול קצר ביותר מ- j מ- i מבחן המסלולים המשמשים בקודדים $\{1, \dots, k\}$. נראה כי G לא מכיל מעגל שלילי אנו יכולים להניח בה"כ כי P מסלול פשוט. ובהכרח מתקיים אחד מהשנאים:

א. אם k אינו קודקוד בינויים ב- P אז המסלול P משתמש רק בקודקודים $\{1, \dots, k-1\}$ ובהכרח אורכו קצר ביותר מבין כל המסלולים המשמשים ב- $\{1, \dots, k\}$ (ואפשר למצוא את אורךו ברכורסיה) ב. אם k הינו קודקוד בינויים ב- P , אז ניתן לחלק את P לשני מסלולים. מסלול p_1 מ- n אל k ומסלול p_2 מ- k אל j . כל אחד מהמסלולים p_1, p_2 לא מכיל את k כקודקוד בינויים (הוא הקצה) ולכן משתמש רק בקודקודים מתוך $\{1, \dots, k-1\}$ כקודודי בינויים. כיון שתת מסלול של מסלול קצר ביותר הוא מסלול קצר ביותר, בהכרח p_1 הוא מסלול קצר ביותר מבין המסלולים המשמשים ב- $\{1, \dots, k-1\}$ מ- n אל k ו- p_2 מסלול קצר ביותר מבין המסלולים המשמשים ב- $\{1, \dots, k-1\}$ מ- k אל j .

לפיכך, נגדיר את המטריצות $D^{(0)}, D^{(1)}, \dots, D^{(n)}$ כך ש- $d_{ij}^{(k)}$ הוא אורך המסלול הקצר ביותר מ- j מבין המסלולים שקודודי הבינויים שלהם מהובוצה $\{1, \dots, k\}$. נגדיר זאת כך:

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & k = 0 \\ \min\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\} & k > 0 \end{cases}$$

אכן הנוסחה מתאימה למה שתיארנו קודם. בדיקן שני אפשרויות לפניינו. להלן האלגוריתם:

Ployd-Warshall(W):

1. $n \rightarrow \text{rows}[W]$
2. $D^{(0)} = W$
3. for $k = 1$ to n
 - a. for $i = 1$ to n
 - b. for $j = 1$ to n

$$d_{ij}^{(k)} = \min\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\}$$
4. return $D^{(n)}$

הסבר: אנו מקבלים פונקציית משקלים ומגדירים את $D^{(0)}$ להיות פונקציית המשקלים עצמה, כלומר כל איבר אם משתמשים רק באיברים מ- 1 עד ... אפס? אין בכלל איברים - לכן בדרך לעבור מ- j באשר לא משתמשים בקודודי בינויים זה בדוק אם קיימת קשת ביןיהם. לאחר מכן רצימ בשילוש לולאות וממש מחשבים את נוסחת הנסיגה. לבסוף כਮובן מחזירים את $D^{(n)}$ שהיא המטריצה שימושתית באיברים $\{1, \dots, n\}$ ($O(|V|^3)$ כמובן).

כיצד מחשבים את π ?

א. תוך כדי החישוב של מטריצת המרחקים ניתן לזכור את הבחירה במטריצות עזר $\pi^{(k)}$ ב. ניתן לבנות את המטריצה π מהמטריצה $D^{(n)}$ בזמן $O(|V|)$. כל תא $\pi_{i,j}$ עליה $O(n^3)$ זמן וסה"כ אכן נגע $O(|V|^3)$. נראה כי $\pi_{i,j} = d_{ix} + w_{xi}$ $\in \{x | d_{ij} = d_{ix} + w_{xi}\}$. סתכל על זוג קודקודים (i, j) . ידוע כי d_{ij} שומר את המרחק בניהם, כלומר את אורך המסלול הקצר ביותר מ- j אל מסלול אחר מותאם הוא כמובן פשוט הקשת ביןיהם. אחרת, אנו יודעים שקיים מסלול אחר קצר יותר. נסמן את הקודקוד הקודם ב- x ומייצגים $d_{ij} = d_{ix} + w(x, j)$ ו- $w(x, j) = d(i, j)$ וכן את הקודקוד הנוכחי ב- y . לכן במקרה הראשון $\pi_{i,j}$ החישוב של π דרוש מעבר על כל הקודקודים והבקרה הנ"ל. סה"כ אכן $O(|V|^3)$ לחישוב π .

5.4 הסגור הטרנסיטיבי של גרף מכובן

עבור גרף מכובן $G = (V, E)$ הסגור הטרנסיטיבי $G^* = (V, E^*)$ הוא גרף על אותו קודקודים כך שם ב- G יש מסלול מקודקוד i לקודקוד j אז ב- G^* ישנה קשת ביןיהם.

$$E^* = \{(u, v) | u \rightsquigarrow_G v\}$$

דרך אחת לחישוב E^* היא להשתמש באלגוריתם של פלייד ורשל, עם פונקציית המשקל $w(e) = 1$ לכל $e \in E$ (וכך אנו מודדים אורך של מסלול ולא צלעות). לבסוף, כל זוג שהמරחק ביןיהם יהיה קטן מאנוסף ניצור לו קשת מתאימה ב- E^* .
 מבחינה פרקטית, ניתן לשפר ולהשתמש במטריצות בוליאניות. כל כניסה של המטריצה יכולה להיות רק אפס או אחד. במקרה זה החישוב יהיה

$$t_{ij}^{(k)} = \begin{cases} (i, j) \in E \vee (i = j) & k = 0 \\ t_{ij}^{(k-1)} \wedge (t_{ik}^{(k-1)} \wedge d_{kj}^{(k-1)}) & k > 0 \end{cases}$$

האלגוריתם מקביל לחלוtin לאלגוריתם לחישוב מרחוקים, השיפור הוא רק בכך שהמחשב צריך לבצע פעולות על ביטים במקום על מספרים.
סה"כ זמן הריצה לחישוב הסגור הטרנזיטיבי הינו $O(|V|^3)$

דרך אחרת לחישוב הסגור הטרנזיטיבי היא להריץ BFS מכל קודקוד ולקבל זמן ריצה $O(|V|(|V| + |E|))$

5.4.1 חישוב הסגור הטרנזיטיבי בזמן $|V|^\omega$

נזכר כי בהרצאה לקחנו את מטריצת השכניםות D^k והעלו בחזקה. אסם כנ, נשים לב כי סגור טרנזיטיבי הוא כל המסלולים באורך k בדיק בגרף. אם כן, נזכיר את מטריצת השכניםות D והעלו בחזקה. זה המטריצה של כל המסלולים באורך k בדיק בגרף. אסם כנ, נשים לב כי סגור טרנזיטיבי הוא כל המסלולים עד לאורך k כלשהו. לכן הרעיון יהיה להסיף אחדות על האלכסון. כך, אם קיים מסלול בין שני קודודים, נוכל תמיד ל选取 בולולה העצמית שבגרף. ופורמלית נסתכל על המטריצה $A \vee I$ באשר I היא מטריצת היחידה, $A = D$. אנחנו יודעים שלאחר $|V|$ פעולות כפולה נקבל את כל המסלולים עד אורך $|V|$. נראה כי סך הכל אם נעלם את המטריצה $(A \vee I)^{|V|}$ בחזקת $|V|$ זה עלה לפחות $O(\log|V|)$ כפולה הזמן לכפוף מטריצות וסה"כ. $O(|V|^\omega \log|V|)$. בעת ש- נתקדם בדרך להוריד את \log .

קלט: גראף מכון $G = (V, E)$
פלט: סגור טרנזיטיבי $G^* = (V, E^*)$

שביל להפוך $\log n$ נניח מה הנחות ליטמיות לכל גראף:
1. G הוא DAG (זמן $O(|E| + |V|)$) ניתן לחשב את G^{SCC} והוא בהכרח DAG , אך אם G לא DAG נוכל לפתור עבורי G^{SCC} , שכן בתוך כל רכיב קשרי אנו יודעים כיצד נראה הסגור קלין (בכל יש קשתות אחדות במטריצה כי יש מסלול)
2. G ממויין טופולוגית (אם לא, ניתן למין טופולוגיה ב- $(|V| + |E|)$).
3. V הוא חזקה שלמה של 2. (ברור כי ניתן להניח כי זאת שכן נוכל להוציאי קודודי דומה - כמו אפסים)

נשים לב כי כיוון שהוא ממויין טופולוגית, זו בהכרח מטריצה משולשית עליונה (מתוח לאלכסון יש אפסים כי לא יתכו קשתות בכיוון זה). נחלק את השאר X, Y, Z באשר X ו- Z מטריצות משולשות עליונות.

$$(A \vee I) = \begin{pmatrix} X & Y \\ 0 & 0 & Z \\ 0 & 0 \end{pmatrix}$$

טענה.

$$(A \vee V)^* = \begin{pmatrix} X^* & X^* \times Y \times Z^* \\ 0 & 0 \\ 0 & 0 & Z^* \end{pmatrix}$$

כלומר, בשביל לחשב את החזקה כל שנדרש הוא לבצע הפרד ומשול - לחשב את החלקים השונים של המטריצה. נקבל הפרד ומשול קלאסי בזמן ריצה: $|V| = n$

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n^\omega) = O(n^\omega)$$

וקיבילנו את זמן הריצה הנדרש.

הערה. מה זה אומר קשת בגרף המקורי באורך Y ? אם דולק שם בית, מדובר בקשת בהכרח שמתחליה בצד X ונגמרה ב- Z . בהכרח - ולכן במטריצה שבזאה, בהכרח כופלים את X^* (מתחלים ב- X), מכופלים ב- Y בשביל לעבור על הקשת בין האזוריים ולבסוף מכופלים ב- Z^* בשביל לסיים ב- Z .

5.5 האלגוריתם של Jhonson

הערה: האלגוריתם יניח שיתכננו קשותות שליליות, אך אין מעגל שלילי. ראיינו כי פלייד וורשל רץ בזמן $O(|V|^3)$ ומטפל במקרה הממושקל ללא מעגלים שליליים אך עם קשותות שליליות. כמו כן ראיינו כי אם ישנו קשותות שליליות אך אין מעגלים שליליים ניתן להריץ בלמן פורד מכל קודקוד ולקבל $O(|V|^2|E|)$ אם בגרף אין קשותות שליליות, דיקסטרה רץ בזמן $O(|V|\log|V| + |E|)$ ואם נרץ דיקסטרה מכל קודקוד נקבל $O(|V|^2\log|V| + |E||V|)$.

מה אם הגרף מכיל קשותות שליליות אך לא מעגל שלילי? נרצה להריץ דיקסטרה מכל קודקוד. אבל, צריך לעשות שינוי בגרף - שלא יוכל קשותות שליליות. אחרת: אי אפשר להשתמש בדיקסטרה. נגידיר פונקציית משקל חדשה \hat{w} שתקיים:
 $\hat{w} : E \rightarrow \mathbb{R}^+$. ב. מסלול p מקודקוד u לו יהיה קצר יותר לפי \hat{w} \iff מסלול p מקודקוד u לו יהיה קצר יותר לפי w .

תהי $V \rightarrow h : h$ פונקציית משקל לקודקודים. נגידיר:

$$\hat{w}(u, v) = w(u, v) + h(u) - h(v)$$

лемה 3. יהיו $G = (V, E)$ גראף מכוכן עם פונקציית משקל $w : E \rightarrow \mathbb{R}$ ותהי $h : V \rightarrow \mathbb{R}$ ולכל $(u, v) \in E$ נגידיר $\hat{w}(u, v) = w(u, v) + h(u) - h(v)$.
הה $P = (v_0, \dots, v_k)$ מסלול מ- v_0 ל- v_k . איזי P מסלול מינימלי ע"פ w אם"מ P מינימלי עבור \hat{w} .
הוכחה: נרצה להוכיח כי מותקאים $\hat{w}(p) = w(p) + h(v_0) - h(v_k)$

$$\hat{w}(p) = \sum_{1 \leq i \leq k} \hat{w}(v_{i-1}, v_i) = \sum_{1 \leq i \leq k} w(v_{i-1}, v_i) + h(v_{i-1}) - h(v_i) =$$

$$\sum_{1 \leq i \leq k} w(v_{i-1}, v_i) + \sum_{1 \leq i \leq k} h(v_{i-1}) - h(v_i) = w(p) + h(v_0) - h(v_k)$$

שכן קיבלונו סכום טלסקופי שמשמעותם. נראה כי פיתוח זה נכון עבור כל מסלול m מ- v_0 ל- v_k .
 ככלומר המשקל של כל מסלול m_0 על v_k ע"פ שהוא בדיק המשקל לפי w בתוספת $(v_k) - h(v_0)$.
 ערך זה אינו תלוי במסלול, שכן קבוע מראש, ולכן בהכרח אם m מינימלי לפי w בבירור מינימלי לפי w ולהיפך.

בפרט, אם m הוא מעגל מתקיים $v_k = v_0$ ואז $w(p) = w(p)$ ולכן אם מעגל הוא שלילי לפי w הוא שלילי לפי w . ננדרש.

מסקנה: אם נחשב את מטריצת המרחקים לפי \hat{w} נוכל לעבור אחד על המטריצה $(O)(|V|^2)$ לחשב את מטריצת המרחקים לפי w . אם כן, כל שנותר הוא למצוא משקל לקודוקדים כך שתהיה תמיד חיובית.

אנו רוצחים שהאלגוריתם יזהה אם יש מעגלים שליליים, ואם אין מעגלים שליליים, ימצא את המסלולים הקצרים ביותר. איזה אלגוריתם אנו מכירים שמסוגל זיהוי מעגלים שליליים, וידוע להזיר תשובה משמעותית אם אין מעגלים שליליים? האלגוריתם של בלמן-פורד! הפלט של אלגוריתם של בלמן-פורד משיקל מושך לכל קודקוד מס' - המורח שלו מוקודק מהמקור. הדבר היחיד שנותר להגדיר הוא מיהו קודקוד המקור לאלגוריתם.

נוסיך לרגר קודקוד חדש s ונוסיך קשת מכוכנת m לכל קודקוד שימושה יהיה אפס.
 פורמלית, נגיד $(E', E) = (V \cup \{s\}, G' = \{E \cup \{(s, u) | u \in V\}\})$ באשר $E' = \{E \cup \{(s, u) | u \in V\}\}$ וכן המוגדרת כך:

$$w'(u, v) = \begin{cases} w(u, v) & (u, v) \in E \\ 0 & u = s \end{cases}$$

כפי שכבר נאמר, הרצת האלגוריתם של בלמן פורד תחשב לכל קודקוד $V \in v$ את המרחק $\delta_{G'}(s, v)$ ולآخر מכון גדייר (v, s) $\delta_{G'}(v) = h(v)$.
למען האינטואיציה: נשים לב כי לכל קודקוד קיים מסלול מסוים l שמשקלו אפס. לכן בהכרח קיים מסלול, אם קיימן קטן יותר אליו בהכרח שהוא במשקל שלילי. נראה כי

$$\hat{w}(u, v) = w(u, v) + h(u) - h(v) = w(u, v) + \delta_{G'}(s, u) - \delta_{G'}(s, v)$$

עלינו להוכיח כי $\hat{w}(u, v) \geq w(u, v)$. נסתכל על המסלול הקצר ביותר ב- G' מ- u אל v . כיון שישנה קשת מה אל v הרי אלו מסתכלים על משקל בקשת זו, שמודדר רק על קשותות שקיימות ע"פ אי שוויון המשולש מתקיים $\delta_{G'}(s, v) \leq \delta_{G'}(s, u) + w(u, v)$ ובהעתרת אגף נקבל $0 \leq w(u, v) + \delta_{G'}(s, u) - \delta_{G'}(s, v) = \hat{w}(u, v)$.

סחה ב' – הוכיחנו כי \hat{w} היא פונקציה שתמיד אי שלילית, וכן היא לשמורת מסלולים קצרים ביותר.

והריין של גנeson ללא ספק עבד.

להלן האלגוריתם:

אלגוריתם 1 ג'ונסון ($G = (V, E)$)

- (א) צור קודקוד חדש s והוסף קשתות במשקל 0 מ- s לכל קודקוד $v \in V$.
- (ב) הרץ את האלגוריתם של בלמן-פורד מקודקוד s .
- (ג) אם האלגוריתם של בלמן-פורד מצא מעגל שלילי -
ו. החזר "מעגל שלילי".
- (ד) אחרת
ו. לכל $u \in V$
א'. שומר $h(v) = \delta(s, v)$
הגדיר לכל $(u, v) \in E$.ii
 $\hat{w}(u, v) = w(u, v) + \delta(s, u) - \delta(s, v)$.iii
נ. מטריצה בגודל $n \times n$ $D = (d_{uv})$.iv
ו. לכל $u \in V$
א'. הרץ את האלגוריתם של דיקסטרה עם הפונקציה \hat{w} מ- s כקודקוד מקור
ושומרו את $\hat{\delta}(u, v)$
ב'. עבור כל $v \in V$
ג. $\hat{\delta}(u, v) + h(v) - h(u) \rightarrow d_{uv}$
ד. החזר את D
-

זמן הריצה: האלגוריתם מרץ בלמן-פורד, ולאחר מכן $|V|$ פעמים מרץ דיקסטרה. סה"כ זמן הריצה $O(|V||E| + |V|^2 \log |V|)$

5.6 האלגוריתם של Seidel

5.6.1 הגדרת הבעיה

קלט: גרף $G = (V, E)$ לא מכון ולא ממושקל.
פלט: $APSP$

נשתמש בהנחה מוקלה - **הגראף G קשור.** מכאן נקבל $|E| \leq |V| - 1$. ניתן להניח זאת כי אם הגראף לא קשור, נוכל לפרק את הבעיה עבורה כל רכיב קשירות בפרט (שכן מחשבים מרחוק מסלול קצר ביותר בין קודקודים.)

כמה זמן לוקח להמיר גראף שמיוצג ע"י רשימת שכנות ליצוג ע"י מטריצת שכנות? $O(|V|^2)$

נניח כי הגראף מיוצג ע"י מטריצת שכנות. דקדימון -

$$A_{u,v} := \begin{cases} 1 & (u, v) \in E \\ 0 & \text{else} \end{cases}$$

המטריצה A סימטרית, כיון ש- G אינו מכון ולכון

5.6.2 כפל מטריצות בוליאני

בכפל מטריצות רגיל מתקיים לפי עדי בן צבי -

:Boolean Metrix Multipcation (BMM) שנקרא

$$C_{ij} = \bigvee_{k=1}^n a_{ik} \wedge b_{kj}$$

כלומר, כאשר מיצגים מטריצות בוליאניות אנחנו משתמשים על המיקום h_{ij} . הוא מכפלה של השורה h במטריצה A והעמודה h_j במטריצה B , מופיע שהיה קיים אינדקס אחד k עבورو הערך h_k בשורה h והערך h_k בעמודה h_j הוא 1, אליו נגיד $c_{ij} = 1$.

נראה כי ניתן לחשב BMM באמצעות FMM : מדו"מ במטריצות בוליאניות, נקבל ערך שונה מאפס אם"מ היה ערך k עבورو 0 $a_{ik}, b_{kj} \neq 0$. מכאן שאם יצא לנו ערך 0 $c_{ij} \neq 0$ נגידו בעט 1, ואם יצא 0 הוא ישאר אפס. עלות BMM תהיה $O(n^\omega)$.

באלגוריתם של סידל, אנחנו משתמשים במטריצה A שהינה מטריצת שכנויות, ונגידר את המטריצה:

$$A' = A^2 \vee A$$

באשר A^2 היא מטריצת השכנויות שמכפלה עצמה, ככפל בוליאני.

נרצה להבין מה המשמעות של A^2 . מכפלה של מטריצת השכנויות עם עצמה: A_{ij}^2 משמעות הדבר: היא שלקחנו את השורה h_i : כל השכנים של v_i , ולקחנו את השורה h_j : כל השכנים של v_j , וחישבנו:

$$A_{ij}^2 = \bigvee_{k=1}^n a_{ik} \wedge a_{kj}$$

כלומר - משתמשים על כל השכנים של i , כל השכנים של j : אם מוצאים k מסוימים שהוא שכן של שניהם, המשמעות היא שיש מסלול באורך 2 בין i לבין j .

טענה: במקומות $1 \leq i, j \leq n$ קיימים מסלולים באורך 2 בין קודקוד i לקודקוד j .

המשמעות היא, שבגרף שמיוצג ע"י A^2 (נתיחס אליה גם כמטריצת שכנויות) יש קשת בין קודקוד i לבין קודקוד j אם ו רק אם יש מסלול באורך 2 בין i לבין j .

נזכיר כי רצינו להגדיר את המטריצה כך $A' = A^2 \vee A$. A' משמעות הדבר היא עבור כל המסלולים באורך 2 או A או A^2 עבור קשותות המקוריות. כאשר נבצע *or* בין המטריצות, במטריצת הפלט A' יש 1 אם"מ בין קודקוד i לבין קודקוד j יש מסלול באורך 1 (קשת) או מסלול באורך 2.

טענה: אם"מ ב- G קיימים מסלולים באורך 1 או 2 בין הקודקודיים i לבין j .

הערה חשובה: $A_{ii}^2 = 1$ אם"מ ל colum i יש שכן שכן משמעות הדבר כי קיים עבورو $i \rightarrow j \rightarrow i$ ווגם $A_{ki} = 1$ אם נסתכל על זוג קודקודיים $i - j$ עם הקשת ביןיהם, נראה כי המסלול $i \rightarrow j \rightarrow i$ הוא גם מסלול באורך 2. שזרר למקומות. משמעות הדבר הינה, שב- A^2 על כל האלכסון יהיה אחידות. ומה זה אומר אם קיימים מסלול באורך 1 מ- i ל- j ? שינוי לולאה עצמאית. ולכן תמיד אנחנו נאפס את האלכסון הראשי של A^2 .

5.6.3 טענות 1, 2

הגדרה: נגידר (u, v) -אורך המסלול הקצר ביותר מ- u ל- v , כאשר G' הוא הגרף המיוצג ע"י A' .

טענה 1: $\delta'(u, v) = \left\lceil \frac{\delta(u, v)}{2} \right\rceil$

איןתוואציה לטענה: ראיינו כי בגרף G' יש קשת בין קודקודים שב G היה בניהם מסלול באורך 2, משמעות הדבר היא שעל כל צלע שהולכים ב' G' צריך ללבת פי 2 בגרף G . כלומר $x' = 2x$ ומכאן $x = \frac{x'}{2}$.

הוכחה: נסנו ב P מסלול קצר ביותר פ"ש לו ב G' .

אם האורך של p הוא $2k$ עבור $k \in \mathbb{N}$ אז קיים מסלול מאורך k בין u לבין v , נסנו P . כיוון שהמסלול P מזלוג על כל קודקוד שי ב P . כלומר אם $P = (v_0, \dots, v_{2k-1})$ אז $(P') = (v_0, v_2, v_4, \dots, v_{2k-2})$. לעומת זאת כיוון להראות כי P' הוא מסלול קצר ביותר בין u ו v . נב"ש כי קיים מסלול $p'' = (v_0, u_0, v_1, u_1, \dots, v_{m-1}u_{m-1})$ אשר $k < p$. כלומר $p'' = (v_0, v_1, \dots, v_{m-1})$ או במסלול $p'' = (v_0, u_0, v_1, u_1, \dots, v_{m-1}u_{m-1})$ אמצעו יוצרים לדג על קווקזו, G לא, כלומר מאמצע מסלול בין u ו v לו G שאורכו $2m < 2k$ סתירה לכך P הוא המסלול הקצר ביותר. מסקנה - P' מסלול קצר ביותר בין u ו v .

ולכן אקו $\delta(u, v) = k = \frac{2k}{2} = \frac{\delta(u, v)}{2}$

ההוכחה עשו מפירהiao זוגי - זוגה מואז.

מסקנה: אם למשל $\delta(u, v) = 4$ או $\delta(u, v) = 7$ או $\delta(u, v) = 8$ נרצה להבין מתי $\delta(u, v)$ יכול להיות זוגי והאם הוא זוגי או אי-זוגי. נרצה לפתח שיטה שתבדוק האם $\delta(u, v)$ הוא זוגי או אי-זוגי.

נסתכל על קודקוד w שהוא שכן של u . כמו כן, ישנו מסלול קצר ביותר בין u ו w . וכן, קיים מסלול קצר ביותר בין u ו w וכי א"ש המשולש: $1 \leq \delta(u, w) + 1 \leq \delta(u, v) + 1$ (נשים לב כי בא השוויון השתמשנו פעמיים). מכאן נחלק לקרים:

א. מקרה ראשון - $\delta(u, v) = \delta(u, w) \bmod 2$: כלומר או שניהם זוגיים, או שניהם אי-זוגיים. במקרה זה אנו רואים שחייב להיות כי $\delta(u, w) = \delta(u, v)$. מדוע? מי שווין המשולש - ראיינו כי ההפרש $|\delta(u, w) - \delta(u, v)|$ ומכאן שההפרש שווה ל-0 או 1. עם זאת, הזוגיות/אי-זוגיות שלהם שווה ולכן לא ניתן שההפרש בניהם הוא אחד. מכאן ההפרש בניהם אפס - כלומר $\delta(u, w) = \delta(u, v)$. כמו כן, זה אומר כי w נמצא על המסלול הקצר ביותר פ"ש לו. כמו כן, מכאן קיבל גם כי $\delta'(u, v) = \delta'(u, w)$.

ב. מקרה שני - $\delta(u, v) \neq \delta(u, w)$ או $\delta(u, v) \neq \delta(u, w)$:

$$\delta'(u, w) = \left\lceil \frac{\delta(u, w)}{2} \right\rceil =_{\delta(u, w) - \text{odd}} \frac{\delta(u, w) + 1}{2} \geq_{\delta(u, v) - 1 \leq \delta(u, w)} \frac{\delta(u, v) - 1 + 1}{2} = \frac{\delta(u, v)}{2} = \delta'(u, v)$$

כלומר - $\delta'(u, w) \geq \delta'(u, v)$

ג. מקרה שלישי - $\delta(u, v) = \delta(u, w)$ או $\delta(u, v) = \delta(u, w)$:

$$\delta'(u, v) = \left\lceil \frac{\delta(u, v)}{2} \right\rceil =_{\delta(u, v) - \text{odd}} \frac{\delta(u, v) + 1}{2} \geq_{\delta(u, v) \geq \delta(u, w) - 1} \frac{\delta(u, w) - 1 + 1}{2} = \frac{\delta(u, w)}{2} = \delta'(u, w)$$

כלומר - $\delta'(u, v) \geq \delta'(u, w)$

מסקנה: אם $\delta(u, v)$ זוגי, אז לכל שכן w של v מתקיים $\delta'(u, w) \geq \delta'(u, v)$ ואם $\delta(u, v)$ אי-זוגי, אז לכל שכן w של v מתקיים $\delta'(u, w) \leq \delta'(u, v)$.

מסקנה מהמסקנה - אם אנחנו יכולים לחשב את כל δ' , אנו יודעים לבדוקיחס גודל-קטן בניהם. מכאן: אנחנו יכולים להכריע האם δ זוגי או שאינו זוגי. פרט לבעה אחת - מה קורה

אם $\delta'(u, w) = \delta'(u, v)$ לשם כך נctrיך להעזר בטענה הבאה.

אנו נמצאים במקרה של $\delta(u, v) \geq \delta(u, w)$ או $\delta(u, w) > \delta(u, v)$. נראה כי אם נסתכל על שכן מסוים מאוד w של v , נצליח להגיע במקרה זה למסקנה מעט אחרת. נסתכל על המסלול הקצר ביותר מ- w לא. נסמן ב- x את השכן של w על המסלול (זה שנמצא קודקוד אחד לפני המסלול הקצר ביותר). מתכונות מסלולים קצרים ביותר (למה 1) מתקיים ב المسلול הקצר ביותר).

$$\delta(u, v) = \delta(u, x) + 1$$

שקל לחלווטין:

$$2k = \delta(u, x) = \delta(u, v) - 1$$

נסמן את הביטוי הנ"ל ב- $2k$. אכן $\delta(u, x) = \delta(u, v) - 1$ במקרה $x = w$ במקרה זה. לפה הגדרה:

$$\delta'(u, x) = \left\lceil \frac{\delta(u, x)}{2} \right\rceil = \left\lceil \frac{2k}{2} \right\rceil = \lceil k \rceil = k$$

$$\delta'(u, v) = \left\lceil \frac{\delta(u, v)}{2} \right\rceil = \left\lceil \frac{2k+1}{2} \right\rceil = \left\lceil k + \frac{1}{2} \right\rceil = k + 1$$

מבחן ניתן לראות כי $\delta'(u, v) < \delta(u, v)$. כמובן - בהינתן שנדע למצוא את השכן של v על המסלול הקצר ביותר מ- w לא. נוכל לדעת כי במקרה בו $\delta(u, v) < \delta(u, w)$ או $\delta(u, w) < \delta(u, v)$ אם נסתכל על אותו שכן כזה, יתקיים $\delta'(u, v) < \delta'(u, x)$ (ולא יתקיים $\delta'(u, x) < \delta'(u, v)$) (נשתמש בזה בהוכחה הבאה).

הרעיון של סידיל היה להסתכל על כל השכנים יחד של v .

טענה 2: $\delta(u, v) \geq \deg(v) \times \delta'(u, v) \iff$ (מדוע אנחנו זוקמים לטענה זו? אנו ידעים להכריע אודות הדרוגה. אם נצליח לחשב (ונצליח, באופן רקורסיבי) את $\sum_{w \in \Gamma(v)}$, נדע להכריע האם $\delta(u, v) \geq \delta'(u, v)$. אם לא - הוא בהכרחizi זוגי.)

הוכחה:

\iff אס $\delta(u, v) \geq \delta'(u, v)$ - או לכל שכן w של v יתקיים $\delta'(u, w) \geq \delta'(u, v)$ (כפי שראינו קוזס לכ), וכן

$$\sum_{w \in \Gamma(v)} \delta'(u, w) \geq \sum_{w \in \Gamma(v)} \delta'(u, v) = \deg(v) \delta'(u, v)$$

\implies נראה קוניתריה פוזיטיב. נניח $\delta(u, v) < \delta'(u, v)$. אז כפי שראינו מעלה במקרה זה לא יש שכן x שעכשו $\delta'(u, x) < \delta'(u, v)$ וכן לכל שכן w של v מתקיים $\delta'(u, w) \leq \delta'(u, v)$

$$\sum_{w \in \Gamma(v)} \delta'(u, w) = \delta'(u, v) + \sum_{w \in \Gamma(v) \setminus \{v\}} \delta'(u, w)$$

החלק הימני של הכיטויו $\delta'(u, v) \geq \delta'(u, v) \times (\deg(v) - 1)$ וכן החלק השמאלי $\delta'(u, v) < \delta'(u, v) \deg(v)$

$$\sum_{w \in \Gamma(v)} \delta'(u, w) = \delta'(u, x) + \sum_{w \in \Gamma(v) \setminus x} \delta'(u, w) < \delta'(u, v) \deg(v)$$

כפי שרצינו להראות.

5.6.4 האלגוריתם

עוד לפני שנדון באלגוריתם – נרצה לראות אלגוריתם גנרי:

ALG1(A)

```

1   if  $A$  is all 1s except for the diagonal
2       return  $A$ 
3   else  $\Delta' \leftarrow \text{ALG1}(A^2 \vee A)$ 
4       for  $u, v \in V$ 
5           if  $\delta(u, v)$  is odd
6                $\delta(u, v) = 2\delta'(u, v) - 1$ 
7           else  $\delta(u, v) = 2\delta'(u, v)$ 
8       return  $\Delta$ 

```

האלגוריתם של סידיל יתבסס על אלגוריתם גנרי זה. אלגוריתם זה פשוט יותר להבנה – האלגוריתם מקבל את מטריצת השכניםות A . האלגוריתם יחזיר מטריצה Δ באשר:

$$\Delta_{i,j} = (\delta(i, j))$$

ראשית, ישנו תנאי עצירה: אם G הוא קליקה, נרצה להציג את A . (מדוע? בקליקה לכל $v \in V$ $\delta(u, v) = 1$ מתקיים, במרקזה זה מטריצת השכניםות של הקליקה הינה 0 באלכסון ובכל שאר המיקומות 1. במצב זה – זה בדיקת מטריצת $APSP$ שנרצה להציג.).
 אחרת, אנטנו נרצה לgesht שוב לאלגוריתם עם $A' = A^2 \vee A$. ערךה, יכנס ל' Δ' . נשים לב כי זה יבוצע בתחילת האלגוריתם שוב ושוב, עד שנתקבל Δ' (עד עבור n כלשהו, באשר הוא קליקה).
 לאחר מכן, נעבור על כל זוג קודוקדים v, u . אם $\delta(u, v) = 1$ הוא אי זוגי, אז $\delta'(u, v) = 2\delta'(u, v) - 1$.
 אחרת, ($\delta(u, v) = 2\delta'(u, v)$ [נובע שישות מטעה 1 שראינו], לבסוף מוחזרים את Δ . הרעיון של סידיל התבסס על כיצד אנחנו מכיריעים אודות הזוגיות של (u, v) בשבייל לבצע מה שעשינו באלגוריתם הגנרי.

כעת, נראה את האלגוריתם של סידיל:

SEIDEL(A)

```

1   if  $A$  is all 1s except for the diagonal
2       return  $A$ 
3   else  $\Delta' \leftarrow \text{SEIDEL}(A^2 \vee A)$ 
4        $M \leftarrow \Delta' \cdot A$ 
5       for  $u, v \in V$ 
6           if  $m_{u,v} < \deg(v)\delta'(u, v)$ 
7                $\delta(u, v) = 2\delta'(u, v) - 1$ 
8           else  $\delta(u, v) = 2\delta'(u, v)$ 
9       return  $\Delta$ 

```

האלגוריתם זהה בתחילת הגרף, ובשורה 6 אנחנו משתמשים בטענה: אם $(m_{u,v} < \deg(v)\delta'(u, v))$ אז אנחנו במרקם האזוג. מכאן, ישירות לפि טענה 2 אפשר להבין בלבד מהו $m_{u,v}$. בשורה 4 אנחנו מגדירים את M : היא מכפלה של מטריצת השכוניות המקורית A עם המטריצה Δ' - נראה כי בנקודה u, u ישנה מכפלה של כל השורה u ב' Δ' עם העמודה u ב' A . נראה כי כיוון שהמכפלה בוליאנית, המכפלה של העמודה והשורה יתנו לנו את סכום כל ה' δ' של הקודקודים שהם שכנים של u . מדוע? השורה u היא שורה שמחזיקה את האיברים $(\delta'(u, v_1), \dots, \delta'(u, v_n))$ מכאן שערך שכזה יכנס למכפלה אם' $m_{u,v}$ הוא שכן של v . מכאן שהמכפלה הנ"ל תניב את $\sum_{w \in \Gamma(v)} \delta'(u, w)$.

$$m_{u,v} = \sum_{w \in \Gamma(v)} \delta'(u, w)$$

ומכאן קיבלו את האלגוריתם שבסיס ישירות על טענה 2 - אם $\sum_{w \in \Gamma(v)} \delta'(u, w) < \deg(v)\delta'(u, v)$ אז $\delta(u, v) = 2\delta'(u, v) - 1$ זוגי ולכנו $\delta(u, v) = 2\delta'(u, v)$ אחרת.

סיכום זמן ריצה:

נשים לב כי ניתן לחשב את דרגות הקודקודים ב- $O(|V|^2)$ זמן באופן טריואלי - נזכיר מערך של קודקודים, עבור כל קודקוד $v \in V$ נعبור על השורה המתאימה במטריצת השכוניות ונסכם את מס' הקודקודים u שקיימת $e = (v, u)$ ביהם. נعبור על $O(|V|)$ קודקודים ובכל פעם שצאו נعبור על $O(|V|)$ קודקודים וסה"כ זמן הריצה יהיה $O(|V|^2)$.

מכאן, שבבירור ניתן לחשב את שורה 6 באלגוריתם. נראה כי שורות 8 – 4 – באלגוריתם עלות $O(|V|^\omega)$ זמן, כיון שמבצעים מעבר על $|V|$ קודקודים ובهم מבצעים פעולות שהינם $O(1)$ וכן כופלים בשתי מטריצות, מה שעולה $O(|V|^\omega)$ זמן. סה"כ $O(|V|^\omega) = O(|V| + |V|^\omega) \geq 2 > 1$ כי $\omega > 1$.

מהי נסחנת הנסיגה של האלגוריתם? متى נגע לתנאי העזירה?

הבחנה: אם אורך המסלול הקצר ביותר G הוא ℓ אז ב' G' אורך המסלול הקצר ביותר מ- ℓ הוא $\lceil \frac{\ell}{2} \rceil$. מכאן, לאחר $\log(|V|)$ קריאות רקורסיביות אורך המסלול הקצר ביותר הגך הוא

קליקה. מדוע? אורך המסלול הקצר ביותר הוא לכל היותר $1 - |V|$ קשווות. אנחנו נרצה לדעת מתיגע לקליקה - כלומר מתי אורך המסלול הקצר ביותר יהיה 1. מכאן ש

$$\frac{|V| - 1}{2^n} = 1 \iff |V| - 1 = 2^n \iff n = \log(|V| - 1) = O(\log|V|)$$

וסה"כ קיבל כי סיבוכיות זמן הריצה היא מס' האיטרציות $(|V| \log|V|)$ כפול הזמן בכל אטרציה $O(|V|^\omega)$ ונקבל

$$O(|V|^\omega \log|V|) < O(|V|^3)$$

הבחנה. אם ידוע כי האורך הכى גדול של מסלול בגרף בין שני קודקודים הוא d , אזי סיבוכיות זמן הריצה של האלגוריתם הינה $O(|V|^\omega \log(d))$.

הבחנה. גם אם נדע אלגוריתם טוב יותר LM , בכל מקרה בכל איטרציה מחשבים את M שהוא לא כפל מטריצותبولיאניות, וכך בכל מקרה זה זמן הריצה.

5.7 A^*

נתון גראף ממוקן וממושקל $G = (V, E)$ וכן שני קודקודים $s, t \in V$. נרצה למצוב מק"ב מס' t . נניח כי $w : E \rightarrow \mathbb{R}^+$.

דייקסטרה ירץ כאן בזמן $O(|V| \log|V| + |E|)$.

נרצה לפטור כרגע בעיה בין s ל- t . נשים לב כי טענו שלא קיים פתרון לעבעיה זו של מקור יחיד באזם יותר טוב $SSSP$. אך אנחנו לא נדבר על המקורה הגורע ביותר. איך נעזר בדיקסטרה? נרצה לעצור באשר t יוצא מהתור כי אז הוכחנו שמצאנו מק"ב מס' t וайו לנו צורך בהמשך הריצה של דיקסטרה.

בעת נציג: הרעיון של A^* הינו רעיון אלגוריתמי ונitin להגדיר אלגוריתמים שפועלים לפי טכניקה A^* , אנחנו לא נראה אלגוריתם ישר שפותר A^* אלא כיצד משתמשים ב- A^* להגדרת בעיות ופתרונות.

נססה לחשב על הכוון הבא - נגדיר את פונקציית הפוטנציאל $\mathbb{R} \rightarrow V$: P . מטרת הפונקציה היא כשלעצמה v קרובה אל t או P קטן יותר. ואז, מה שנעשה באלגוריתם של דיקסטרה יהיה להוציא לפחות $d[v] = p(v) + p(u)$ כלומר $d[v] = d[u] + p(u)$, וכך אנחנו בתחלת הריצה של דיקסטרה נתעדף את הצעדים שקרובים אל t . נשים לב - כל הטעינה הנ"ל פוגעת בכל הכנות של דיקסטרה. וכך נשתמש בזה.

נגדיר פונקציית משקל חדשה:

$$w'(u, v) = w(u, v) + p(v) - p(u)$$

נראה כי

$$w'(P_{u \rightarrow v}) = w(P_{u \rightarrow v}) + p(v) - p(u)$$

ולכן כל המסלולים בין u ל- v עם אותה התוסף $p(v) - p(u)$ וכל הסדר בהם נשאר, ובפרט למק"ב.

עם זאת, יש המון בעיות שאנו זוקקים לטפל בהם - בין צמותים שונים (למשל u ול- w) לא מובטח שנשמר הסדר. כמו כן, בשביל שדייקסטרה יעבור נרצה כי $w'(u, v) \geq 0$.

הגדלה: אם לאחר ההתמרה (הגדרת w') כל $0 \leq w' \leq p$ (פונקציית הפוטנציאל) הינה פיזיבלית. כלומר לכל $(u, v) \in E$ $w'(u, v) \geq 0$

лемה 1: אם p פיזיבלית, ויהי $t \in V$ כך ש $\forall v \in V$, $\exists p(t) \leq p(v) \leq \delta(v, t)$

הוכחה: יהי מסלול v_0, v_1, \dots, v_k, t .

$$0 \leq w'(P_{v \rightarrow t}) = w(P_{v \rightarrow t}) + p(t) - p(v) \leq w(P_{v \rightarrow t}) - p(v)$$

וסה"כ נקבל $p(v) \leq \delta(v, t)$. בפרט, זה נכון עבור המסלול הקצר ביותר לכלומר $(P_{v \rightarrow t})$.

מה קורה באשר $p(v) = \delta(v, t)$ לכל קודקוד?

$$p_{s \rightarrow t} = (s, v_1, \dots, v_k, t)$$

נסתכל על (v_i, v_{i+1}) במסלול.

$$w'(v_i, v_{i+1}) = w(v_i, v_{i+1}) + \delta(v_{i+1}, t) - \delta(v_i, t) = \delta(v_i, t) - \delta(v_i, t) = 0$$

משום ש $\delta(v_i, t) = \delta(v_{i+1}, t) + \delta(v_{i+1}, t)$. וכך - כל קשת במרק"ב מ- t ל- s תהיה במשקל 0. נשים לב - כיוון שכל המשקלים על המסלול הנ"ל הינם אפס, דייקסטרה ישיר ירוץ על המסלול שלו מ- s ל- t . הוא ראשית יוציא את s עם $d[s] = 0$ ואז יבצע סדרת הקלות קודם כל על המק"ב שלו מ- s ל- t . מכאן, שהשאיפה שלו היא למצוא פונקציית פוטנציאל שמקربת את $p(v)$ כמה שיותר אל $\delta(v, t)$. שכן אנחנו לא יודעים את $\delta(v, t)$ שכנ שביב לדעת אותו צריך להריץ דייקסטרה, ואנחנו לא מעוניינים לעשות זאת. נשים לב כי את $\delta(v, t)$ נוכל למצוא אם נריץ דייקסטרה על G^T מ- t .

מסקנה מהחישוב: אם נמצא ערך $\delta(v, t)$ קרוב מאוד ל- $p(v)$ אז ערך הקשת יהיה קרוב יותר לאפס.

лемה 2: אם p פיזיבלי ו-0 אי ניתן להפוך ל- p' פיזיבלי עם $p'(t) \leq 0$

הוכחה:

נגדיר

$$p'(v) = p(v) - p(t)$$

ברור כי עבור $0 \leq p'(v) \leq p(v)$. כל שניות להראות הוא כי p' פיזיבלי. תהי $(u, v) \in E$. אז,

$$w''(u, v) = w(u, v) + p'(v) - p'(u) = w(u, v) + p(v) - p(t) - p(u) + p(t) = w'(u, v) \geq 0$$

שכן מראש הנחנו כי p פיזיבלי ולכן $w'(u, v) \geq 0$.

דוגמה ראשונה לשימוש ב^{*} A :

נרצה לבדוק מרחק קצר ביותר בין שני צמתים במשור. אם נתן לשכן צמתים במישור, נסתכל על מפה דו מימדית וצומת (x_1, y_1) וצומת (x_2, y_2) . נסמן $\|g(u) - g(v)\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.
 אם סתכל על פונקציית פוטנציאלית $p(t) = \|g(t) - g(t)\|$ אז $p(v) = \|g(v) - g(t)\| = \|g(v) - g(t)\|$ מכיוון $0 \leq p(t) \leq p(v)$ ונראה כי היא פיזibilית.

$$w'(u, v) = w(u, v) + p(v) - p(u) = w(u, v) + \|g(v) - g(t)\| - \|g(u) - g(t)\|$$

נראה כי אם נדרוש $w(u, v) \geq \|g(u) - g(v)\|$ ברור שהכביש יהיה אורך יותר מהמרחק האוירי.
 נניח שזה מותקאים, אז

$$\geq \|g(u) - g(v)\| + \|g(v) - g(t)\| - \|g(u) - g(t)\| \geq 0$$

באשר המעבר האחרון מאי שוויון המשולש על מרחקים אוקלידיים.

דוגמה שנייה לשימוש ב^{*} A : נניח צומת מרכזי f וידוע $\delta(v, f)$ מכל $v \in V$ אליו. (כלומר כמו זמן לוקח להגיע מכל קודקוד ליעד מרכזי). נראה כי אין כן מביצז נתונים ומפעיל דיקסטרה אל f מכל קודקוד אך רוצה להשתמש בתנוז זה לדעת יותר - למשל, בהינתן המידע הזה, לאיזה צומת אני מעוניין להגיע מהמיוקם הנוכחי שלי, כך ששה'כ המרחק ממנה ל f יהיה קצר ביותר.
 בהינתן מידע זה נרצה להגיד את הפונקציה הבאה:

$$p(v) = \delta(v, f) - \delta(t, f)$$

אכן $0 \leq p(t) \leq p(v)$, נוכיח פיזibil:

$$w'(u, v) = w(u, v) + p(v) - p(u) = w(u, v) + \delta(v, f) - \delta(t, f) + \delta(t, f) - \delta(u, f) =$$

$$w(u, v) + \delta(v, f) - \delta(u, f) \geq 0$$

שכן המעבר נובע אוטומטית מאי שוויון המשולש.

מסקנה: ניתן להריץ את האלגוריתם של דיקסטרה אם מגדירים לו פונקציית פוטנציאלית טובה, פיזibilית, ולקבל שבאופן ישיר האלגוריתם הראשית יבצע את המסלול שאינו מעוניין בו, מה שיוריד את זמן הריצה.

5.7.1 הגדרה פורמלית

קלט: גראף מכובן $G = (V, E)$ עם פונקציית משקל $p : V \rightarrow \mathbb{R}^+$ ופונקציית פוטנציאל $w : E \rightarrow \mathbb{R}$ (פונקציית הערךות של המרחק $\delta(\cdot)$) ושני קודקודים s, t .

פלט: מסלול קצר ביותר מס s ל- t .

טענה. אם לכל $V \in u \in p(u) = \delta(u, t)$ אז סריקת A^* תקבע רק בקודקודים על גבי מסלול קצר ביותר מס s ל- t .

הוכחה. נזכר כי המשקל של כל קשת הוא $w(u, v) = w(u, v) + p(v) - p(u)$, לכל קודקוד u $\hat{\delta}(s, u) = \delta(s, u) + p(u) - p(s) = \delta(s, u) + \delta(u, t) - \delta(s, t)$ (המרחק לפי \hat{w}) שכן אנו יודעים שאכן u על המסלול. משמע - כל קשת שנמצאת על המסלול הקצר ביותר עלותה אפס.

אם כן, לכל קודקוד v שאינו במק"ב מתקיים $\hat{\delta}(s, v) = \delta(s, v) + p(v) - p(s) = \delta(s, v) - \delta(s, t) > \delta(s, t) - \delta(s, v) + \delta(v, t) > \delta(s, t)$ שכן v אינו על המסלול הקצר ביותר. מסקנה - דיקסטרה יקבע קודם בקודקודים שימושיים אפס, וכך בהכרח יעבור ראשית על המסלול שלנו מס t ויתעדך אותו. שכן דיקסטרה מעתה לא יכולה אל s .

נשים לב - לא נבע מטענה זו שזמן הריצה יהיה O של אורך המסלול, שכן יתכן מצב אחד בדיק בעייתי: אם כל הקשות בגרף הם אפסים, במקורה זהה דיקסטרה לא בהכרח יתעדך את הקודקודים על המסלול שלנו. לכן במקרה הגורע ביותר נגיעה לנו לזמן הריצה של דיקסטרה. אם כן - הטענה כן אומרת שלא נקבע בקודקוד שלא במק"ב.

הגדרה. תהי פונקציית פוטנציאל $\mathbb{R} \rightarrow V : p$, $p(u) \leq \delta(u, t)$ אם תקרא קבילה אם $\forall u \in V$

טענה. אם לכל $V \in u \in p(u) \leq \delta(u, t)$ אז סריקת A^* קבילה - כלומר $\delta(s, u) + p(u) > \delta(s, t)$ (כלומר ישנו שביקר בקודקוד לא טוב - אבל לא נקבע בקודקוד ממש לא טוב).

הוכחה.

לכל קודקוד u במסלול קצר ביותר מס t מתקיים כי $\hat{\delta}(s, u) = \delta(s, u) + p(u) - p(s) \leq \delta(s, u) + \delta(u, t) - p(s) = \delta(s, t) - p(s)$

אם כן, לכל קודקוד v שאינו במק"ב. נב"ש שביקרנו בקודקוד v שאינו במק"ב שימושיים $\hat{\delta}(s, v) + p(v) > \delta(s, t)$.

$$\hat{\delta}(s, v) = \delta(s, v) + p(v) - p(s) > \delta(s, t) - p(s)$$

כלומר, כתע קיבלו קודקוד שגדל מעריך $\delta(s, t) - p(s)$ אך ראיינו שקודקודים על המק"ב הם לכל היותר ערך זה ולכן אין סיבה שנבחן בו. בסתיויה.

טענה. אם לכל $V \in u \in p(u) \leq 2\delta(u, t)$ אז סריקת A^* לא תקבע בקודקוד $V \in u$ כך $\delta(s, u) + p(u) > 2\delta(s, t)$.

הוכחה.

לכל קודקוד u במסלול קצר ביותר מס t מתקיים כי $\hat{\delta}(s, u) = \delta(s, u) + p(u) - p(s) \leq \delta(s, u) + 2\delta(u, t) - p(s) = 2\delta(s, t) - p(s)$

אם כן, לכל קודקוד v שאינו במק"ב. נב"ש שביקרנו בקודקוד v שאינו במק"ב שימושיים $\hat{\delta}(s, v) + p(v) > 2\delta(s, t)$.

$$\delta(\hat{s}, v) = \delta(s, v) + p(v) - p(s) > 2\delta(s, t) - p(s)$$

כלומר, כתע קיבלנו קוודוק שגדול מערך $(s - p, t)$ אך ראיינו שקוודוקים על המק"ב הם לכל היותר בערך זה ולכון אין סיבה שנבקר בו. בסתיו.

6 הרצתה 6: רשתות זרימה (Network Flow)

מעט מוטיבציה: נניח ואנחנו חברה שרצה להעביר נפט נוזלי מנוקודה A לנוקודה B . בינו מראש רשת של צינורות שמאפשרות העברת שכבזו. בצד אחד אפשר להכנס את הנפט מצד אחד והוא יכול לצאת מזו הצד השני. לכל צינור, ישנה קיבולת אחרת. וכן: לכל צינור ישנו קווטר שונה, קווטר גדול יותר מאשר מאפשר להזרים יותר נפט דרך הצינור. אפשר לדמיין את הצינור כקשת בגרף מכון. היא יכולה לנوع בדיק בכוון אחד. מקור הנפט, בנינו מערך של צינורות שונים. כמו כן, ניתן שעוברים שני צינורות בין שתי נקודות: אחד לכל כיוון. המטריה שלנו היא להעביר כמה שיטור נפט מקודוק התחלה s אל קוודוק היעד t . בכל צינור, אפשר להעביר עד מס' ליטרים מסוימים. המטריה שלנו היא בהינתן רשת הצינורות, לחשב כמה "פט" ניתן להזרים בשנייה ברשת. נשים לב כי המושג להזים לא הוגדר היטב.

נשים לב כי בהינתן צינור c_{n1} מ- t ל- $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4 \rightarrow s$, אם בכל הצינורות אפשר להזרים מהה, אך בצד אחד $a_4 \rightarrow a_3$ אפשר להזרים רק 2 למשל, זה לא עוזר לי: אני נתקע מאחר עם 98 ליטר נפט. מערכת ה"ביבוב" תתקע, אי אפשר לצבור במיקום מסוים נפט/, מים שיצטברו. לכן אסור מראש להעביר שם(!) מהה. נשים לב שצריך מראש לדעת מה כמות הליטר המקסימלית שמותר לנו להעביר במסלול, שלא יוצר מצב של להתקע.

נשים לב כי תחנן למשל רשת זרימה $t \rightarrow a_1 \rightarrow a_2 \xrightarrow{a_2 \rightarrow a_3 \rightarrow a_4} s$ מסוג זה, באשר נקודת a_2 היא נקודת פיצול. נניח שכל הצינורות בגודל קיבולות 100, אך $a_2 \rightarrow a_3 \rightarrow a_4$ בקיבולות 2, וכן $a_2 \rightarrow a_5$ בקיבולות 50. במצב זה, נוכל להעביר 2 ליטר בצינורות דרך t ל- $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4 \rightarrow s$. וסה"כ הקיבולות ברשת הזרימה תהיא וונכל להעביר עד 50 דרכ $t \rightarrow a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4 \rightarrow s$. $50 + 2 = 52$

חידד: אין כאן משמעות לזמן, אלא בהינתן רשת זרימה, כמה אפשר להעביר בה בכל מצב שהוא. וכן: כמות ה"פט" שנכנס אל קוודוק ברשת הזרימה s שווה לכמות ה"פט" שיצא מהקוודוק t . המיקום היחיד שמן יכול להיווצר נפט הוא s , והמקום היחיד שיוכל להשתאר בו / ליצאת נפט: קוודוק t .

דוגמאות לשימוש: להבין מהו קצב העברת המידע האפשרי בין שני מחשבים ברשת מוחשבים בזמן העברת קוובץ ענק בין המחשבים. דוגמה נוספת היא לחשב איזה מסילות רכבות ניתן להפיצו בעלות הקטינה ביותר על מנת למנוע מעבר של ציוד מנוקודה אחת לשנייה.

6.1 הגדרה פורמלית של זרימה

הגדרה: רשת זרימה היא גרפ' מכון $G = (V, E)$ עם קוודוק מקור $s \in V$ וקוודוק יעד $t \in V$ ופונקציית קיבולת $C : V \times V \rightarrow \mathbb{R}^+$ $\cup \{0\}$ באשר $C(u, v) > 0 \iff C(v, u) = 0$.

בבהרה. הקיבולות יכולות להיות מס' ממשי חיובי או אפס. היא אפס אם אין קשת בין קוודוקדים, אחרת: היא גדולה ממש מאפס. נפרמל,

$$C(u, v) = \begin{cases} x \in \mathbb{R}^+ & (u, v) \in E \\ 0 & (u, v) \notin E \end{cases}$$

נتابון בשתי הגדרות, שקולות עבור **זרימה**. ההגדרה הראשונה יותר אינטואטיבית, והשנייה פחותה (אך תעזר לנו בהמשך עם המתמטיקה).

הגדרה ראשונה: זרימה ברשות זרימה $G = (V, E)$ עם פונקציית קיבול f , היא פונקציה : $f : \{0\} \cup V \times V \rightarrow \mathbb{R}^+$ שמקיימת את התנאים הבאים (נדגיש - פונקציית הזרימה היא מה שאנו מארימים בפועל על הרשות) :

1. **אלוצי קיבולות:**

$$\forall u, v \in V : 0 \leq f(u, v) \leq c(u, v)$$

כלומר, בהכרח הזרימה תהיה גודלה-שווה מאפס (לפי הגדרת הקיבולות, אם אין קשת היא אפס). וכן הזרימה לא יכולה לעבור מעולם את הקיבולות (כי לא יכול לעולם להעביר את הזרימה דרך הרשות).

2. **שיעור זרימה:** סכום הזרימה שנכנס שווה לסכום הזרימה שיצא. חוק שימור הזרימה.

$$\forall v \in V \setminus \{s, t\} : \sum_{u \in V} f(u, v) = \sum_{u \in V} f(v, u)$$

הערך של זרימה f הוא:

$$|f| = \sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$$

כלומר, סכום כל ערכי הזרימה של הקודקודים שיצאים מ- s , פחות כל ערכי הזרימה שנכנסים אל s . נשים לב כי ניתן לזרום חזרה אל s זרם. טה"כ ערך זה הערך שיצא מ- s , בניקוי מה שזר. ככלומר: משוש הסכום נטו שיצא לבסוף.

הגדרה שנייה: זרימה ברשות זרימה $G = (V, E)$ עם פונקציית קיבול C , היא פונקציה \mathbb{R} שמקיימת את התנאים הבאים:

1. **אלוצי קיבולות:**

$$\forall u, v \in V : f(u, v) \leq c(u, v)$$

נשים לב כי בהגדרה זו, יתכן כי הזרם יהיה שלילי. הוא רק לא יכול לעבור את הקיבולות. זה מאד מוזר מבחינה מתמטית: ההסבר לכך יהיה הסימטריה מטה.

2. **סימטריה:**

$$\forall u, v \in V : f(u, v) = -f(v, u)$$

נשים לב שגם הגדירה מאוד אינטואטיבית וחשובה. אם עברו לנו מצד מסויים 6 יחידות, מצד החפוך אליו עבר 6-. באופן דומה: אם מישחו הביא לי 100 שקל, אצלי עלה 100 שקל ואצלו ירד 100 שקל.

3. שימור זרימה: בהגדרה זו אנחנו מסתכלים רק על הקשותות שיווצאות מקודקוד מסוים, ונאמר שסכום הזרימה שלם הוא אפס.

$$\forall v \in V \setminus \{s, t\} : \sum_{u \in V} f(v, u) = 0$$

הסביר: אנו מעוניינים כי אם נכנס מצד אחד של s סכום זרם של 12 למשל, מהכוון שנכנס אל s מצד השני של הקודקוד יזרום סכום זרם של -12. (וכן כמובן שבסוף יתרכז שזרמו מאותו צד של הקודקוד סכום שהתאפס לאפס. בכל מקרה: מדובר בקשנותות שנכנסות אל s). נשים לב כי עדיין ישנו שימור זרימה כמו בהגדרה הקודמת, אבל מהגדרת הסימטריה צריך ליעזג זאת מתמטית קצת אחרת. נתן לומר כי אם מסתכלים על כמה שיוצא, אם מצד אחד יוצא 15 ערך זרם, נראה שנכנס אליו גם -15 (אך בכיוון השני מופיע שנכנס, יצא 15 – בגלל סימטריות) וכך אכן $0 = 15 - 15$.
ערך של זרימה f הוא:

$$|f| = \sum_{v \in V} f(s, v)$$

כעת, ערך הזרימה יוגדר להיות כל מה שיוצא s , ושוב נשים לב שהוא שקול אינטואטיבית לבעה הקודמות, מושימטריה, אם נכנס חזרה 5 ייחזרת אל s זה אליו יצא 5 – (וזה הסימטריה). לכן אם יצא s 15 למשל, ונכנס 2. זה שקול לכך שיוצא 15 מ- s ויצא 2 – (מסימטריה) ולכן ערך הזרימה הוא $15 - 2 = 13$.

בקורס נשתמש רק בהגדרה השנייה. הגדרה הראשונה לטובת אינטואיציה בלבד.

הערה. נשים לב כי הפונקציה $f = 0$ היא גם פונקציית זרימה.

6.2 הגדרת הבעה

קלט: רשת זרימה $G = (V, E)$ עם פונקציית קיבול $\{0\} \cup \mathbb{R}^+$.

פלט: זרימה f ב- G בעלת ערך גדול ביותר מבין כל הזרימות האפשריות. *Max flow*

6.3 תכונות של זרימה

נרצה להרחיב את ההגדרה של זרימה לקבוצות.

הגדרה: יהיו $X, Y \subseteq V$. נגדיר את הזרימה בין שתי קבוצות הקודקודים כך:

$$f(X, Y) = \sum_{x \in X} \sum_{y \in Y} f(x, y)$$

טענה: $f(\{s\}, V) = |f|$
הוכחה:

$$f(\{s\}, V) = f(s, V) = \sum_{v \in V} f(s, v) = |f|$$

טענה 4: תהי f זרימה בראשת זרימה $G = (V, E)$.
אי. $\forall X \subseteq V : f(X, X) = 0$.1
אי. $\forall X, Y \subseteq V : f(X, Y) = -f(Y, X)$.2

מתקיים: $\forall X, Y, Z \subseteq V \wedge X \cap Y = \emptyset$.3
 $f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$.a
 $f(Z, X \cup Y) = f(Z, X) + f(Z, Y)$.b

הוכחה:
 תהי f רשות זרימה. ויהי $X, Y, Z \subseteq V$.
 .1

$$f(X, X) = \sum_{x \in X} \sum_{x_2 \in x} f(x, x_2) = \sum_{x \in X} 0 = 0$$

.2

$$f(X, Y) = \sum_{x \in X} \sum_{y \in Y} f(x, y) = \sum_{x \in X} \sum_{y \in Y} -f(y, x) = -\sum_{y \in Y} \sum_{x \in X} f(y, x) = -f(Y, X)$$

.3. נניח כי $X \cap Y = \emptyset$

$$f(X \cup Y, Z) = \sum_{w \in X \cup Y} \sum_{z \in Z} f(w, z) =_{X \cap Y = \emptyset} \sum_{x \in X} \sum_{z \in Z} f(x, z) + \sum_{y \in Y} \sum_{z \in Z} f(y, z) = f(X, Z) + f(Y, Z)$$

лемה 5: תהי f זרימה ברשות זרימה $G = (V, E)$. אזי

$$|f| = f(V, t)$$

הוכחה:

$$f(V, V \setminus \{s, t\}) = -f(V \setminus \{s, t\}, V) = -\sum_{x \in V \setminus \{s, t\}} \sum_{y \in V} f(x, y) = -\sum_{x \in V \setminus \{s, t\}} 0 = 0$$

כמו כן נשים לב כי

$$V = \{s, t\} \cup (V \setminus \{s, t\})$$

וכו שתי קבוצות אלו זרות.icut לפיה טענה 4 מוגדר כי:

$$|f| = f(s, V) = f(V, V) - f(V \setminus \{s\}, V) = 0 - f(V \setminus \{s\}, V) = f(V, V \setminus \{s\})$$

$$= f(V, V \setminus \{s, t\}) + f(V, t) = 0 + f(V, t)$$

ושה"י **ყילוי** $f(V, t) = |f|$ כנדרש.

מסקנה: ראיינו כי $|f| = f(s, V) = f(V, t) = |f|$ ומלמה 5 ראיינו כי $f(s, V) = f(V, t)$ ונקבל כי f קלומר: סך הזרם שיוציא מ- s שווה לזרם שוכנס אל- t .

הגדרה: חתך (s, t) הוא חתך $(S, T) = (S, V \setminus S)$ כאשר $s \in S$ וכן $t \in T$. נשים לב כי G הינו מכון ולכן קשת שחותча את החתך היא קשת שעוברת M אל T (קשתות בכיוון השני לא נקראות כאלו ולא מעניינות אותנו). כמו כן בהכרח $\emptyset \neq S \subset V$.

למה 6: יהיו $G = (V, E)$ רשת זרימה עם פונקציית קיבולות C . ותהי f זרימה ב- G . יהיו (S, T) חתך של G , אז $|f| = f(S, T)$.

$$|f| = f(S, T)$$

כלומר, אם נסתכל על הזרימה משמאלי S אל T , סכום הזירומות הללו הוא בדיקע ערך הזרימה).

הוכחה:
נשים לב כי $T \cap S = \emptyset$ וכן $S \cup T = V$. כמו כן

$$f(S \setminus \{s\}, V) = \sum_{x \in S \setminus \{s\}} \sum_{y \in V} f(x, y) =_{(*)} 0$$

כיוון ש $x \neq s$, לפי חוק שימוש הזרימה (3).

$$f(S, T) = f(S, V) - f(S, S) = f(S, V) - 0 = f(S, V) = f(S \setminus \{s\}, V) + f(s, V) = 0 + f(s, V) = |f|$$

כנדרש.

הגדרה: קיבולות בין קבוצות הינה

$$C(S, T) = \sum_{x \in S} \sum_{y \in T} c(x, y)$$

$$\text{למה 7. יהיה } f(S, T) \leq C(S, T).$$

הוכחה:

$$f(S, T) = \sum_{x \in S} \sum_{y \in T} f(x, y) \leq \sum_{x \in S} \sum_{y \in T} c(x, y) = C(S, T)$$

מסקנה. לכל חתך גודל הזרימה זהה לפי למה 6 ולכל זרימה בחתך חסומה ע"י הקיבול של החתך, לכן כל זרימה חסומה ע"י כל החתכים ובפרט הקטן ביותר, ובפרט הזרימה הגדולה. לכן ערך הזרימה המקסימלי, יהיה בהכרח קטן שווה מהקיבול הקטן ביותר. $\text{Max flow} \leq \text{Min cut}$, **כלומר**,

6.4 שיטת פורץ-פלקרים

נניח שאחנו מתחילהים $f = \sum_{v,u \in V} f(u,v) = 0$ (כפי שהערכנו קודם קודם זו אכן זרימה). קלומר: $\forall v, u \in V$

$$\text{מכן } |f| = 0.$$

נניח שאחנו מסתכלים על רשת זרימה $G = (V, E)$ ומאננו מסלול כלשהו בין s ל t . איז, ברגע מסוון בזרימה המקסימלית האפשרית באותו המסלול: הוא ערך הזרימה המינימלי שמוספע על המסלול. קלומר אם מאננו מסלול $t \rightarrow_{14} \rightarrow_{50} a_2 \rightarrow_{100} a_1 \rightarrow s$ אז הזרימה המקסימלית האפשרית במסלול הינה 14.

נרצה להגדיר פונקציית זרימה כך

שיי מסלול P . אם $(x, y) \in P$ היא קשת עם $C(x, y) = \min_{(u, v) \in P} \{C(u, v)\}$

$$f'(u, v) = \begin{cases} C(x, y) & (u, v) \in P \\ -C(x, y) & (v, u) \in P \\ 0 & o.w \end{cases}$$

$$|f'| = C(x, y)$$

וכן נשים לב שאכן ערך הזרימה הינו $C(x, y)$ כי זה בדיקת 14 עליו דיברנו קודם בדוגמה. ערך הקיבולות הקטן ביותר על המסלול הינו ערך הזרימה.

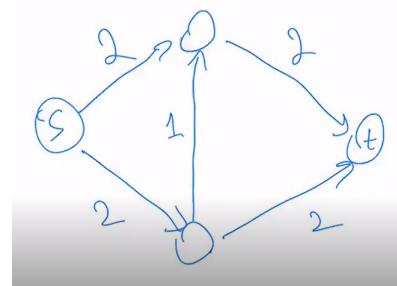
נראה כי אכן שיפרנו את הזרימה מ-0 ל $C(x, y)$.

יתכן כי ישנים הרבה מסלולים זרים (בקשותות) מ-0 ואז אפשר להפעיל את הרעיון שעשינו קודם על כל המסלולים. אם כן, זה לא מכסה את כל המקרים. באופן כללי להיות שהיינו רוצחים להשתמש בקשת אחת עבור יותר ממסלול אחד (נקודות פיצול למשל). לשם כך צריך לפתח מנגנון שיאפשר לקחת את המינימלי בכל מסלול, אך להארים יותר במידעה שנייתן להתפצל לאורך המסלול.

הגדרה: קשת $(x, y) \in E$ תקרא רוויה תחת זרימה f אם $f(x, y) = C(x, y)$ או אין רוויה, אז ניתן להשתמש בקבולות הנורטרות.

הערה: כל עוד קיים מסלול מ- s לששתותוי איינו רווות, "נארים" על מסלול זה "כמה שאפשר". נשים לב שהרעיון הוא בגדר רעיון ולא מוגדר היטב. עם זאת: זה לא יעבוד.

נסתכל על הדוגמה הבאה:



נראה כי נרצה ללקת במסלול שבצורת Z מ- s מטה, עובר ב-1 ומסתיים ב- t . הערך המינימלי במסלול זה הינו 1. וקיים לנו כי הקשת 1 רוויה. המסלול היחיד שנותר לנו מ- s לששתותוי אין רווות זה להתחיל מ-1, ללקת על המסלול של שתי הקשות שערוך 2. נשים לב שעל מסלול זה לנצל להזרים

1 בלבד כי הקשת 2 שמניעה מלבילה אל t זורם בה כבר 1 (מהמסלול הקודם). מכאן שנסתכל על המסלול האחרון שלא כל הקשתות בו ורויות, המסלול שמתחל ב s מטה וועבר רק בקשתות שערכו 2. שוב: הקשת 2 שמנעה מלטיה אל t השתמשה באחד ולכן ניתן להזרים במסלול זה. 1. סה"כ האזרים 1 בכל מסלול והיו לנו 3 מסלולים וקיים כי ערך הזרימה הינו 3 $= |f|$. עם זאת: היה ניתן להזרים 4 במעבר ישיר של 2 מט מעלה ומטה. מסקנה: הרעיון לא טוב. והבעיה - הרבה יותר קשה משחצבנו.

6.5 הרשות השירית

נשים לב כי הבעיה בדוגמה שהראנו קודם, היא שהאלגוריתם קודם בחר את המסלול שעובר דרך 1. אם הוא לא היה בוחר במסלול זה, או היה מנסה לבחור אותו אחרון: הפתרון כן היה עובד באשר לדוגמה הספציפית הקודמת. נראה כי אלגוריתם חמדן בוחר החלטה ואחר כך חייב לעמוד בה, הוא לא יכול להתרשם. במקרה של קודם, הינו שמחים אם לאחר הבחירה במסלול שעובר ב1 הוא היה יכול להתחרט. מכאן נגיעה להגדרה הבאה.

הגדרה: هي $G = (V, E)$ רשות זרימה עם פונקציית קיבולת C . ותהי f זרימה ב G . הקיבולת השירית של C_f תחת f היא הפונקציה

$$C_f(u, v) = C(u, v) - f(u, v)$$

כלומר: כמה עוד יש לנו להזרים בקשת מסויימת.

הגדרה: הרשות השירית של f תחת G היא רשות זרימה $G_f = (V, E_f)$ שפונקציית הקיבולת שלה הינה C_f . וכך $\{(u, v) | C_f(u, v) > 0\} = E_f$. נשים לב כי אכן פונקציית הקיבולת מקיימת 0 $\leq C_f(u, v) \leq C(u, v)$ תמיד $C(u, v) \geq f(u, v)$.

כלומר: קבוצת הקשתות זה כל הקשתות שעדו ניתן להזרים בהן.

נשים לב כי אמרנו שתתכן זרימה שלילית. מכאן: אם בכיוון $t \rightarrow x$ זורם 2, ולא הייתה קשת בכיוון ההפוך. ככלומר 2 $f(x, y) = 2$, נראה כי בכיוון השני לא עברה זרימה ולכן $0 = C(y, x) = c_f(y, x) = C(y, x) - f(y, x) = f(x, y) = 2$. במקורה שלנו, בכיוון ההפוך יירום אותו ערך ברשות השירית. ומכאן המשקנה: ניתן כי ברשות השירית יתוכנו קשתות נוספות היו בגרף המקורי.

נראה כי כתוצאה מהרשות השירית, בעת פיתחנו מגננו ל”זורה אחוריה” במידה ולא מעוניינים במה שבחרנו. בעת יש את האפשרות ללקת בכיוון הנגיד ולחשוף מסלול שישלים. מתי נדע לעצור? כאשר מסלול מס L : ככל הקשתות נכנסות מט ולא יוצאות ממנו. הרעיון יהיה לשפר מסלולים על הרשות השירית עד שלא ניתן יהיה לעשות זאת.

הגדרה: בהינתן מסלול P ברשות השירית G_f נגדיר (את הקיבות השירית המינימלית) כך:

$$C_f(P) = \min\{c_f(u, v) | (u, v) \in P\}$$

6.6 שיטת פורד-פלקריםון

להלן האלגוריתם:

FORD-FULKERSON($G = (V, E)$, s, t, c)

- 1 initialize $f(u, v) = 0$ for all $u, v \in V$
- 2 $G_f \leftarrow G$, $c_f \leftarrow c$
- 3 **while** there exists a path P from s to t in G_f
- 4 $c_f(P) \leftarrow \min_{(u,v) \in P} \{c_f(u, v)\}$
- 5 **for** each edge $(u, v) \in P$
- 6 $f(u, v) \leftarrow f(u, v) + c_f(P)$
- 7 $c_f(u, v) \leftarrow c_f(u, v) - c_f(P)$
- 8 $f(v, u) \leftarrow f(v, u) - c_f(P)$
- 9 $c_f(v, u) \leftarrow c_f(v, u) + c_f(P)$
- 10 update E_f
- 11 Return f

מה קורה באלגוריתם? האלגוריתם מקבל פונקציית קיבולות, רשת זרימה וקודקוד מקור ויעד. בתחילת: האלגוריתם מאתחל את פונקציית הזרימה להוות אפס עבור כל הקודקודים. כמו כן: מתחילה את רשת הזרימה השורית להוות דומה לזרימת עצמה ואת הקיבולת השורית להוות הקיבולת. לאחר מכן נכנים אל לולאה שמתבצעת כל עוד קיים מסלול מ s ל t ברשת השורית. מגדירים את (P) כפְיַהוּגֶדֶר לעיל, עוברים על כל זוג קודקודים במסלול P , מוסיפים $f(v, u)$ את $c_f(P)$ שגילינו קודם וכן מכמה בעית אפשר לעברו (v) ובאותו זמנה מוריידים אותו מ(u) וכן מגדירים את הרשת השורית באופן דומה ונגיד: מ(v) אנו מוריידים את $c_f(P)$ כי בעית יש שם פחות זרם שנitin להעביר) ואל $c_f(u)$ אנחנו מוסיפים את $c_f(P)$ כי יש יותר זרם שנitin להעביר). לבסוף: מעדכנים את הקשות E_f (יתכן שיש קשותות בעית שימושיים או לחלוין מוריידים). ככלומר כל מי שהקיבולת השורית שלו התאפשרה צריך להעיף, מי שקדם לנו היה אפס וכעת לא: צריך להכניסו לרשת השורית. פעולה זו היא למעשה העדכון של G_f .

הגדרה: במסלול P אנחנו נקראו "מסלול שיפור". וכן אנחנו משתמשים ב P בשביל לשפר את f .

6.7 נוכנות האלגוריתם וזמן הריצה

נתבונן בבעיית חתך מינימום:

קלט: גרף $G = (V, E)$ מכובן. ושני קודקודים $s, t \in V$ וכן פונקציית משקל $w : E \rightarrow \mathbb{R}^+$
 פלט: חתך (s, t) שסכום קשותות שעוברות מ- T הוא כמה שיוטר קטן.
 בשביל לפטור את בעיית זרימת המינימום נרצה לפטור בעיה של זרימה ברשת שנגיד, ועל מנת לראות שזה אכן פטור את הבעיה נוכיח את נוכנות האלגוריתם של פורד (ואז כבר קיילנו את הנוכנות שרצינו).

למה 7: תהי $G = (V, E)$ רשת זרימה עם פונקציית קיבולות C ותהי f רשת זרימה ב- G . תהי $|f| \leq C(S, T)$ חתך (s, t) ב- G . אז, (S, T) כלומר, ערך הזרימה בגרף יהיה קטר-שווה מסכום הקיבולות של הקשתות שחווצות את החתך (T) משמאלי S אל ימין T .

הוכחה: תהי $G = (V, E)$ רשת זרימה עם פונקציית קיבולות C ותהי f רשת זרימה ב- G . תהי $|f| = f(S, T)$ ראיו נבר כי $|f| = f(S, T)$ בлемה 6. לכן

$$|f| = f(S, T) = \sum_{x \in S} \sum_{y \in T} f(x, y) \leq_{(*)} \sum_{x \in S} \sum_{y \in T} C(x, y) = C(S, T)$$

באשר (*) זה כיוון שתמייז מתקיים כי $f(x, y) \leq C(x, y)$, כלומר הזרימה היא לכל היותר בגודל הקיבולות. נדרש.

סימנו: נסמן את זרימות המקסימום $|f^*|$.
מסקנה: ערך כל זרימה שהיא $|f|$ יהיה קטן או שווה לחטך (s, t) המינימלי.

6.7.1 max-flow - min-cut

משפט 8: תהי $G = (V, E)$ רשת זרימה עם פונקציית קיבולות C ותהי f זרימה ב- G . אז, כל התנאים הבאים שקולים:
 א. f זרימת מקסימום.
 ב. ב- G_f אין מסלול שיפור.
 ג. קיים חתך (s, t) שנסמן ב- G כך ש $|f| = C(S, T)$.

מסקנה: זרימת מקסימום = חתך (S, T) מינימום (!)
מסקנה שנייה: המשפט מוכיח את נכונות האלגוריתם, כיוון שא' גורר את ב' באם'ם. אכן אם אין מסלול שיפור מצאנו את זרימת המקסימום.

הוכחה:
 א \iff ב: נניח כי f זרימת מקסימום. נניח בשיילה כי f זרימת מקסימום וב- G_f יש מסלול שיפור. מכאו, ניתן להשתמש ב- P על מנת להגדיל את ערך הזרימה ולכון f אינה זרימת מקסימום, בסתיו.
 ב \iff ג: נניח כי G_f און מסלול שיפור. נגדיר את (S, T) נזקלטן:

$$S = \{v \in G_f \mid \exists P = (s, \dots, v)\}$$

$$T = \{v \in G_f \mid \text{not } \exists P = (s, \dots, v)\}$$

כלומר S היה קבוצת הקזקוזדים שקיים מסלול מעליו, ו- T זו הקבוצה שלא קיים מסלול מעליו.
 נראה כי אכן קיים מסלול מעליו s לא s ולכון $s \in S$ ולא קיים מסלול מעליו t כי און מסלולי שיפור ולכון $t \in T$. ולכן אכן חתך (s, t) שהוגדר חתך (S, T) .

טעינה 6: לכל $u \in S$ ו- $v \in T$ מתקיים $f(u, v) = C(u, v)$

הוכחה: מצד אחד תמי' מתקיים $f(u, v) < C(u, v)$. מצד שני כשלילה כי $f(u, v) \leq C(u, v)$. נניח כשלילה כי $C_f(u, v) = c(u, v) - f(u, v) > 0$ כלות,

$$C_f(u, v) = c(u, v) - f(u, v) > 0$$

ומכאן קיילו כי $f \in E_f(v, u)$. ע"פ הגדרת S , יש מסלול מ- v ל- G_f . מסלול זה ייחד עם הקשת $f \in E_f(v, u)$ יוצר מסלול מ- v ל- G_f בסתיו רק ש- $f \notin E_f$ icut,

$$|f| = f(S, T) =_{def} \sum_{x \in S} \sum_{y \in T} f(x, y) =_{Lemma 9} \sum_{x \in S} \sum_{y \in T} C(x, y) = C(S, T)$$

ג \iff א: נניח שקיים חטץ (s, t) שיסמכו G כז ש- $|f| = C(S, T)$. אך, נניח כשלילה כי f אינה זרימת מקסימום. ככלומר, קיימת פונקציית זרימה f' כז ש- $|f'| > |f|$. מכאו שבחרכו $f'(S, T) \leq C(S, T)$ כי $|f'| > |f| = C(S, T)$ ש

כదרש.

6.7.2 סיבוכיות זמן הריצה (פורד פרלקסון)

באופן כללי, השיטה של פורד פרלקסון עלולה שלא להסתיים לעולם. עם זאת, אם כל הקיבולות הם מספירים שלמים: האלגוריתם כן יסתתיים.

מכאן נובע, שבכל איטרציה הזרימה תשתרף בפחות אחת. וכך, אם הזרימה המקסימלית הינה

$|f^*|$ אי לכל היותר לאחר $|f^*|$ איטרציות האלגוריתם סיימיים.
נראה כי בכל איטרציה אנו נדרשים למצוא מסלול - למשל באמצעות dfs . זה יעלה $O(|E_f|)$ וכן מתבצעים עדכונים על המסלול שעלוותם $O(|V|)$ סה"כ כל איטרציה עולה $O(|E| + |V|)$. הנחיה: נניח כי כל הקודודים ב- V נמצאים על מסלול כלשהו מס' t בגרף המקורי. אחרת, אפשר בזמן לינארי להוציא את אלו שלא נמצאים ונקבל מכאן כי $|E| \leq |V| - 1$ ולכן כל איטרציה עלותה $O(|E| + |V|) \leq O(|E| + |V|)$.
מכאן נקבל כי זמן הריצה הינו: $(|f^*| \times |E|)$

лемה 10: תהי רשת זרימה $G = (V, E)$ עם פונקציית קיבולות C וזרימה f , אי מותקים f' זרימה ב- G $f + f' \iff G_f \neq G_{f'}$ זרימה ב-
מסקנה: f' זרימת מקסימום ב- G $f + f' \iff G_f = G_{f'}$ זרימת מקסימום ב- G .

נשים לב, גם אם כל הקיבולים הם רצינליים ניתן להגדיר את זמן הריצה הנ"ל שכן ניתן בתחליה למצוא את המכנה המשותף ולהכפיל בו, להריץ אלנו רגיל על מס' שלמים ובסוף חרך חזרה במכנה המשותף.

6.8 מציאת חטץ (s, t) מינימום בראשת זרימה

נשים לב, שלפי משפט MFMC אם נרץ אלגוריתם לזרימת מקסימום, נדע את הגודל של החטץ (s, t) מינימום, אך לא נדע כיצד למצוא חטץ שכזה. אם כן, נרצה למצוא אותו.

בהתanton רשת זרימה $G = (V, E)$, c, s, t נרצה להחזיר חטץ $(S, V \setminus S)$ כז שהחטץ הוא (s, t) מינימום.

לפיכך, נתבונן באלגוריתם הבא:
 $:find-min-cut(G = (V, E), c, s, t)$

- א. הרץ אלגוריתם למציאת זרימת מקסימום בראשת הזרימה ותהי f זרימת המקסימום המותקבלת.
- ב. חשב את הרשת השירית G_f
- ג. חשב את הקבוצה $\{u \in V \mid \exists s \sim u\}$ כלומר כל הקודקודים שקיים מסלול מ- s אליהם בראשת השירית.
- ד. החזר את החתק $(S_f, V \setminus S_f)$

נרצה להוכיח נכונות.

- א. נרצה להראות שאכן מוחזר חתק (s, t)
- ב. החתק הוא (s, t) מינימום.

טענה 2. תהי $(G = (V, E), c, s, t)$ רשת זרימה וכי $(S_f, V \setminus S_f)$ החתק שהוזר כפלט מהרצת האלגוריתם. אזי, הוא חתק (s, t) .

הוכחה. לפי הגדרת הקבוצה בהכרח $S \in s$ כיון שישנו מסלול מס' t ל- s כמו כן, לפי משפט $MFMC$ אכן כיון שהזרימה מקסימלית לא קיים מסלול שיפור כלומר מסלול מס' t ולכן $t \in S \setminus V_f$.

טענה 3. תהי $(G = (V, E), c, s, t)$ רשת זרימה וכי $(S_f, V \setminus S_f)$ החתק שהוזר כפלט מהרצת האלגוריתם. אזי, הוא חתק מינימום.

הוכחה. נרצה להוכיח שככל קשת אשר חוצה את החתק, אכן רווחה. כלומר $(v, u) = f(u, v) = c(u, v)$ תהיה קשת (v, u) כך ש- $v \notin S_f$, $u \in S_f$. נניח בשלילה כי הקשת איננה רווחה. כלומר בהכרח $c(u, v) < c(u, v) - f(u, v) > 0$. מהגדלת S_f זה גורר כי קיים מסלול מקודקוד s אל v כי $c(u, v) - f(u, v) > 0$ ולכן $v \in S_f$. ככלומר קיבלנו כי קיים מסלול מס' t אל u ומהם אל v מעבר על הקשת: ולכן $v \in S_f$ בסתייה.

אם כן,icut נוכיח כי כל קשת (v, u) לא מובילה לזרימה. נב"ש שקיימת קשת (u, v) עבורו $0 > f(u, v) > 0$. כלומר בהכרח $v \in S_f$. ושוב ניתן להציג כי סך הזרימה שיוצאת מהקבוצה $S_f = S_f + c(S_f, V \setminus S_f)$ שווה ל- s . סה"כ שילוב שתי הטענות נקבע כי סך הזרימה שיוצאת מהקבוצה S_f הוא שווה לערך הזרימה. שימור הזרימה, כיון ש- $s \in S_f$ ש- s ערך זה שווה לסך הזרימה שיוצאת מקודקוד s כלומר לערך הזרימה.

זמן ריצה: האלגוריתם מחשב זרימת מקסימום ומבצע פעולה לינארית לחישוב הרשת השירית והקבוצה S_f , למשל ע"י הרצת BFS , ולכן סה"כ עלות חישוב זמן ריצה האלגוריתם הוא כחישוב זרימת מקסימום.

6.9 הכרעה האם זרימת מקסימום \ חתק מינימום ייחודיים

כפי שראינו אנו מסולים למציאת חתק מינימלי (s, t) בראשת הזרימה וגם זרימת מקסימום.

קלט: רשת זרימה $(G = (V, E), c, s, t)$

פלט: האם קיים חתק (s, t) מינימום יחיד בראשת הזרימה.

לפיכך, נתבונן באלגוריתם הבא:

- א. הרץ אלגוריתם למציאת זרימת מקסימום בראשת הזרימה ותהי f זרימת המקסימום המותקבלת.
- ב. חשב את הקבוצה $\{u \in V \mid \exists s \sim u\}$
- ג. חשב את הקבוצה $\{u \in V \mid \exists u \sim t\}$
- ד. החזר שקיים חתק ייחד אם $S_f \cup T_f = V$

טענה 9. יהי (S, T) חתק מינימום בראשת הזרימה, אזי כל קשת (u, v) עבורו $c(u, v) = f(u, v)$ מקיימת $c(u, v) = f(u, v)$ הוכחה. נסתכל על הזרימה שהוצאה את החתק, מתקיים

$$|f| = \sum_{u \in S} \sum_{v \in T} f(u, v)$$

$$\sum_{u \in S} \sum_{v \in T} f(u, v) \leq \sum_{u \in S} \sum_{v \in T} c(u, v) = C(S, T)$$

אבל בכלל שמדובר בחalkץ מינימום מתקיים $|f| = C(S, T)$ וכך נקבל כי

$$\sum_{u \in S} \sum_{v \in T} f(u, v) = \sum_{u \in S} \sum_{v \in T} c(u, v)$$

ולכן לכל קשת החוצה את החalkץ נקבל $f(u, v) = c(u, v)$

טענה 10. יהי (S, T) חalkץ מינימום כלשהו, אז $S_f \subseteq S$ והוא בשלילה כי קיים $s \in S_f$ ו- $v \in T$ אשר לא קיים מסלול מ- s אל v וכן קיון $t \in T$ אשר לא קיים מסלול בין s ו- t . סה"כ קיים מסלול $t \sim u \sim s$ וסה"כ קיים מסלול שיפור ברשות השיוורית בסטירה $MFMC$.

טענה 11. יהי (S, T) חalkץ מינימום כלשהו, אז $T_f \subseteq T$

טענה 12. קיים חalkץ מינימום ייחיד אם $S_f \cup T_f = V$ והוכח. הוכיחנו שלכל חalkץ מינימום (S, T) מתקיים $S_f \subseteq S, T_f \subseteq T$ ולכן אם קיימים שני חalkצי מינימום שונים $(S, T), (S', T')$ נקבע סטירה לעובדה ש- $S_f \cup T_f = V$, ונניח בשלילה שקיימים שני חalkצי שונים $(S, T), (S', T')$. ואם נרצה לפרמל, נניח כי $S \subseteq S' \cup T_f = V$ ו- $v \in S$ ו- $v \notin S'$. מהשונות, קיים קודקוד u כך ש- $u \in S \wedge v \in T$. אם $u \in S_f$ נקבע כי $S_f \subseteq S'$ ולכן $v \in S$ בסטירה. ואם $u \in T_f$, נקבע כי $v \in T$ אך קיבלו סה"כ $v \in S \cap T = \emptyset$.

(לטובת האינטואיציה, זה אומר שכל קודקוד ידוע בדיקת באיזה צד הוא נמצא בגרף השיוורי. לכן בהכרח חalkץ מינימום ייחיד).

7 הרצאה 7: זרימה - אדמוניס קארפ ו-Karp

ראינו בהרצאה הקודמת את השיטה של פורד למציאת רשת זרימה בעלות של $O(|f^*| \times |E|)$ אם הקיבולות מספרים שלמים. אלגוריתם נוסף שנראה כתה הוא אדמוניס קארפ שרך בסיבוכיות זמן $O(|V| \times |E|^2)$. ולאחר מכן נראה אלגוריתם של *Dinic* שרך בסיבוכיות זמן $O(|V|^2 \times |E|)$.

7.1 האלגוריתם של אדמוניס קארפ

האלגוריתם של אדמוניס קארפ הוא צורת מימוש לשיטה של פורד-פרקלטון. בכל שלב אנחנו נמצאים במסלול שיפור ברשות השיוורית בעל מספר מינימלי של קשתות, מציאת המסלול מתבצעת על ידי הרצת *BFS* מ- s עד למינימום t . בזרור כי כל שיפור מסלול עולתו $O(|V| + |E|) = O(|V| + |E|)$ כי מינימום קשירות, ולאחר שוכניך כי האלגוריתם מוצא את זרימת המקסימום בתחום לכל היותר ($|V| \times |E|$).

איורציות נקבל כי סיבוכיות זמן הריצה של האלגוריתם הינה $O(|E|^2 \times |V|)$.

אלגוריתם 1 אדמונדס-קארפ ($G = (V, E), s, t, c$)

1. לכל קשת $e \in E$

$$0 \rightarrow f[u, v] \rightarrow f[v, u] \quad (\text{א})$$

2. כל עוד קיים מסלול ברשות השירית G_f מ- s ל- t .

(א) הרץ BFS מ- s עד מציאת t . ויהי p המסלול שנמצא בעז המסלולים הקצריים ביותר מ- s ל- t .

$$\min \{c_f(u, v) \mid (u, v) \in p\} \rightarrow c_f(p) \quad (\text{ב})$$

לכל קשת $e \in p$

$$f[u, v] - c_f(p) \rightarrow f[u, v] \quad (\text{i})$$

$$-f[u, v] \rightarrow f[v, u] \quad (\text{ii})$$

האלגוריתם משתמש בReLU של פורד וממשו אותו שונה. נרצה להוכיח נכונותו.

הגדעה: נסמן $\delta_f(v, u)$ באורך המסלול הקצר ביותר בין u ל- v ב- G_f .

лемה 11: תהיו f' זרימה המתknבלת מזרימה f ע"י שיפור על גבי מסלול נארוך הקצר ביותר מ- s ל- t . אזו לכל $u \in V$ ו- s מתקיים $\delta_{f'}(s, u) \leq \delta_f(s, u)$ (כלומר, כמסלול השיכור קצוויזים רק מתקיים מקוזוקה המקורי).

הוכחה: נניח בשילhouette שהוא לא המיצג. ככלומר קיים קוזוקה V ו- u המקיימים $\delta_{f'}(s, u) > \delta_f(s, u)$. יתנו פס' קצוויזים נילובים v והוא קוזוקה במרחב מיינימלי מ- s ב- $G_{f'}$ שערכו זה מתקיים. והוא מסלול קצר ביותר מ- s ל- v לאחר השיפור, ככלומר קוזוקה מיינימלי מ- s ב- $G_{f'}$. והוא קוזוקה הקוץ מ- s ב- P . מתקיים $\delta_{f'}(s, v) \leq \delta_f(s, v)$ וכן $\delta_{f'}(s, u) = \delta_f(s, v) + 1$. נולך לפזרים -

א. אם $f(v, u) < c(v, u)$ אז היחס $f(v, u) < c(v, u)$ גוייתם גם ב- G_f . וכך

$$\delta_f(s, u) \leq \delta_f(s, v) + 1 \leq \delta_{f'}(s, v) + 1 = \delta_{f'}(s, u)$$

בסתירה לכך $\delta_{f'}(s, u) > \delta_{f'}(s, v)$.

ב. אם $f(v, u) = c(v, u)$ או $v \notin E_f$ היחס $f(v, u) = c(v, u)$ כולם בהכרח השיפור שעשו ערך בקשר (v, u) בכיוון הפוך ל- s, v . כיוון שהשיפור נעשה על פיו מסלול $'p$ שהוא קצר יותר מ- s שלו הוא קצר ביותר ובפרט היחס $f(v, u)$ הוא על מסלול קצר ביותר מ- s ל- u ב- G_f . וכך

$$\delta_f(s, u) = \delta_f(s, v) - 1 \leq \delta_{f'}(s, v) = \delta_{f'}(s, u) - 1 < \delta_{f'}(s, u)$$

ושוב בסתירה להנחה כי $\delta_{f'}(s, u) > \delta_{f'}(s, v)$.

מסקנה 2: תהיו f' זרימה המתknבלת מזרימה f על ידי שיפור על גבי מסלול נארוך קצר ביותר מ- s ל- t . אזו, לכל $u \in V$ ו- s מתקיים $\delta_{f'}(u, t) \leq \delta_f(u, t)$ (כלומר, אם יש שוויון שכזה אזו לאחר השיפור לא יוכלו מסלולים קצרים יותר חזשים).

лемה 13: תהיו f' זרימה המתknבלת מזרימה f ע"י שיפור על גבי מסלול נארוך קצר ביותר מ- s ל- t . אז $\delta_{f'}(s, t) = \delta_f(s, t)$. אם $\delta_{f'}(s, t) = \delta_f(s, t)$ אזו כל מסלול קצר ביותר מ- s ל- t ב- $G_{f'}$ הוא גם מסלול קצר ביותר מ- s ל- t ב- G_f .

הוכחה: נגידו מושג חזש של קשותות חדשות בגרף - קשת (u, v) היא קשת חדשה אם ורק אם (u, v) הייתה ב- G_f ומסלול השיפור כליל אותה. כיוון שמסלול השיפור הוא מסלול מאורך קצר ביחס למסלול נובטח כי $\delta_f(s, u) + 1 \geq \delta_f(s, v)$. היות P מסלול קצר ביחס למסלול t ב- $G_{f'}$. נניח בשיילה כי P לא מעכז ב- G_f . אז P בהכרח מכיל קשת חדשה (ויתכו שיתור מעתה). תהיו (u, v) קשת חדשה שהיא למעשה P . לפיה $\delta_{f'}(s, v) \geq \delta_f(s, v) + 1$ מתקיים $\delta_{f'}(s, v) \geq \delta_f(s, v) + 1$ ולפי משקנה 2 מקבל כי (t, u) מתקיים $\delta_{f'}(u, t) \geq \delta_f(u, t) + 1$. כיוון שמסלול קצר ביחס למסלול t ב- $G_{f'}$ ($u, v \in P$) אז

$$\delta_{f'}(s, t) = |P| = \delta_{f'}(s, v) + 1 + \delta_{f'}(u, t)$$

מצד שני כיוון (u, v) היא קשת על מסלול השיפור שהוא מסלול מאורך קצר ביחס למסלול G_f מתקיים כי $1 \leq \delta_f(s, v) = \delta_f(s, u) + 1$. כלומר:

$$\delta_{f'}(s, t) = |P| = \delta_{f'}(s, v) + 1 + \delta_{f'}(u, t) \geq$$

$$\delta_f(s, v) + 1 + \delta_f(u, t) = \delta_f(s, u) + 1 + 1 + \delta_f(u, t) =$$

$$\delta_f(s, t) + 2 > \delta_f(s, t)$$

וקיילנו $(t, u) > \delta_f(s, t)$ בסתירה לכך שהם שווים.

נשים לב כי כאשר הזרימה f' מתקבלת מזרימה f ע"י שיפור של מסלול שיפור P_f מאורך קצר ביחסו, או מסלול זה P_f לא יכול להוות קיטס גם ב- $G_{f'}$ כיוון שלפחות אחת מקשתויה הייתה רוויה. תובנה זו מובילה לлемה הבאה - שתאפשר לנו את זמנה הרווחה של אדמינים קארוף:

лемה 14: תהיו G רשת זרימה f וזרימה כלשהי. נתבונן באלגוריתם אשר משפר על גבי מסלולים מאורך קצר ביחסו מ- f רק כל עוד אורכם הוא (t, u) או, מרגע שקשת (v, u) הייתה רוויה ע"י האלגוריתם, קשת זאת לא תהיה בשימוש על ידי אף מסלול שיפור אחר בימהלך ריצת האלגוריתם. (הוכחה זהה לлемה 13).

כעת, נסתכל על ריצת האלגוריתם אדמינים קארוף, ונסתכל על כל מסלול השיפור שאורכם ℓ . נקרא לאיסוטוויות שיפורו אותן: הפאהזה E_ℓ של האלגוריתם. ככלומר: הפאהזה E_ℓ באלגוריתם של E_k היא סדרת האיטרציות שכזו אוריך המסלול הקצר ביחסו הוא ℓ קשותות. פאהזה יכולה להיות יקרה. כל מסלול שיפור בפאהזה E_ℓ ינוו לשייך לקשת אחת (לפחות) אותה הוא הפך לרוויה. לפי הלמה, מובטח שכל קשת תשייך למסלול אחד בפאהזה E_ℓ לכל היותר. ולכן סה"כ בפאהזה E_ℓ יכולות להיות לכל היותר $|E|$ איטרציות.

מסקנה 5: יש לכל היותר $|E|$ איטרציות בכל פאהזה. (בכל איטרציה בפאהזה משפרים לפחות אחת, ולפי לemma 14 לא משתמשים בה שוב ולכו לכל היותר במקורה הגורע ישנים $|E|$ איטרציות פר פאהזה).

בנוסף, כיוון שיש לכל היותר $1 - |V|$ מרחוקים אפשריים של s או, מס' הפאזות הינו $O(|V|)$. מכיוון שהי' מס' האיסטרוציות של האלגוריתם היו לכל היותר $O(|E| \times |V|)$. כמו כן, כל איטרציה זורשת הריצת BFS כדי שכביר אפורי, שעילוגה $O(|E|)$ ניקבל את סיבוכיות ימ"ו הריצה: $O(|E|^2|V|)$.

נשים לב כי הנכונות של פורד פלקרטון נכונה גם כאן, שכן תמיד יוכל להריץ במקביל את האלגוריתם זהה ואת האלגוריתם הרגיל ולקחת את המינימום מביניהם.

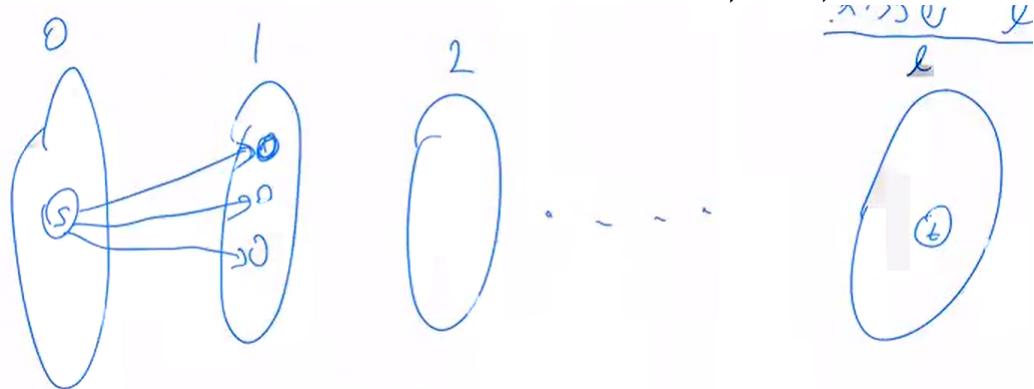
7.2 גרען השכבות

באלגוריתם של דינץ' יש עדין $O(|V|)$ פאות. כל פאה תעללה סה"כ $O(|E| \times |V|)$ זמן. ואז זמן הריצה יהיה $O(|V|^2|E|)$.

הגדרה: יהיו $G = (V, E)$ גרף מכון עם קודקוד מקור s וקודקודיעד t , נסמן ב- ℓ את $\delta(s, t)$ (מס' הקשנות במסלול הקצר ביותר). נגדיר באמצעותו את גרף השכבות, בו יש $\ell + 1$ שכבות, ובשכבה ה- ℓ יהיו כל הקודקודים $u \in V$ כך ש- $\delta_f(s, u) = \ell$ וגם u נמצא על מסלול קצר ביותר מס' אל t . (כלומר, הוא Tat מסלול של מסלול קצר ביותר).

סה"כ, קיבל את השכבות $\ell = 0, 1, 2, \dots$. נשים לב כי בהכרח בשכבה 0 ישנו את s ורק את s . כמו כן, בהכרח בשכבה ℓ ישנו את קודקוד t ורק אותו שכן $\delta(s, t) = \ell$ לפי הגדרה. שכן, נב"ש שישנו קודקוד אחר $x \in \ell$ שאינו ארוך המסלול הקצר ביותר מס' אל x הוא ℓ וכן אם הוא שacle חלק מסלול קצר ביותר מס' אל t , בסתיו כי המסלול הקצר ביותר שהוא חל בו אל t הוא באורך $\ell + 1$, ואז בהכרח זה גורר קיום שכבה $\ell + 1$ בסתיו.

כך נראה גרף שכבות:



הקודקודים שנמצאים אל ℓ הם קודקודים מהשכבה $1 - \ell$ שנמצאים על המסלול הקצר ביותר בדרך אל t באורך $1 - \ell$ ואמנם נסיך את הקשת שאיתה הם נמצאים אל ℓ או נקבל מסלול קצר ביותר באורך $\ell = \delta(s, t)$.

הקשות בין השכבות הן קשותות שנמצאות על אייזחו מסלול קצר ביותר מס' אל t . נשים לב כי לא כל הקודקודים נמצאים בגרף השכבות – רק קודקודים שנמצאים על קשותות שנמצאות על אחד מן המסלולים הקצרים ביותר מס' אל t . באופן דומה, לא כל הקשותות נמצאות בגרף השכבות אלא רק הקשותות שבאחד המסלולים הקצרים ביותר.

צומת u נמצא בשכבה j באשר $\delta_f(u, t) = i - j \iff 0 \leq j \leq i$ וכן קשת $v \in L_{i,j+1}$ באשר $u \in L_{i,j}$ ומן $v \in L_{i,j+1}$ באשר (u, v)

הרעין באלגוריתם של דינץ', יהיה בתחילת כל פaza לבנות גוף שכבות L מ- G_f .

זכור מהי פaza - הסתכלנו על ריצת האלגוריתם אדמונס קארפ, וכן הסתכלנו על כל מסלולי השיפור שאורכם ℓ (יתכנו כמה כאלה). נראה לאטרציה ששירפו אותנו: הפaza ה- ℓ של האלגוריתם. ככלומר: הפaza ה- ℓ באלגוריתם של Ek היא סדרת האטרציות שבוחן אורך המסלול הקצר ביותר והוא ℓ קשיות. פaza יכולה להיות ריקה. ובקיצור: **בכל אטרציה אנחנו מוצאים מסלול מגדיל באמצעות BFS** המסלול הקצר ביותר יש לו אורך d , **כל עוד האורך הזה לא משתנה** - אנחנו באותה פaza ברגע שהאורך גדול $d \rightarrow d+1$ - פaza חדשה מתחילה.

לפי מה, 4, שבתחלת כל פaza, ב- G_f נמצאים כל המסלולים הקצרים ביותר שהאלגוריתם ימצא תוך כדי הפaza. לכן, לפי ההגדרה של L : גוף השכבות L מכיל את כל המסלולים הקצרים ביותר מס- t אל t ב- G_f .

בתחלתינו אנו בונים את גוף השכבות, לוקחים את גוף השכבות. כמה עולה לבנות את גוף השכבות? **זמן הבניה של גוף השכבות הוא** $O(|V| + |E|)$ - כי יש קשרות. כיצד? מרכיבים BFS מס, ומרכיבים BFS מ- t על G^T .icut, לפי עונה שקת (u, v) נמצאת במסלול קצר יותר אמר'ם $\delta(s, t) = \delta(s, u) + 1 + \delta(u, t) = \delta(s, u) + \delta(u, v)$, נוכל לבדוק לכל $(u, v) \in E$ האם מתקיים השוויון $\delta(s, t) = \delta(s, u) + \delta(u, v)$ ואם כן היא על מסלול קצר יותר מס- t ונוסף אותה לגוף השכבות. סה"כ בניית גוף השכבות עולה $O(|E| + |V|)$ אך $O(|E| + |V|) = O(|E|)$.

7.3 מציאת מסלול

כיצד מוצאים מסלול קצר ביותר עזרת גוף השכבות? מתחילהemos שבחבבה 0, ואנו יודעים כי כל קשת מס- t טוביל אותו לקודקוד שנמצא בשכבה הראשונה. בדומה, בשכבה 1 לא משנה איזה קשת נבחר לעבור לקודקוד בשכבה השנייה. באופן כללי, אם אנו בשכבה ה- i , ישנו קודקוד s_i , אנו יודעים כי גם ישנו הרבה קשות מהשכבה ה- i לשכבה $i+1$, לא משנה איזה קשת נבחר היא תמיד נמצאת על מסלול קצר ביותר כלשהו מס- t . לפי הבניה של L , כל קשת $e \in L$ נמצאת על מסלול קצר יותר מס- t . לכן, בחירה של קשת שרירותית שיווצאת מוקודקוד s בשכבה ה- i בהכרח טוביל לקודקוד שנמצא בשכבה ה- $i+1$.

לכט, האלגוריתם למציאת מסלול קצר ביותר כלשהו מס- t מואוד פשוט:

- נתחל את $s \rightarrow u$. כל עוד $t \neq u$ בחר קשת שרירותית שיווצאת מס- s , (u, v)
- נוסיף את (v, u) למסלול
- ועדכן $u = v$ ו חוזר לשלב א'.
- לבסוף, החזר את המסלול.

כמה זמן לוקח למציאו מסלול קצר ביותר מס- t ? שכח? נראה כי אנחנו בוחרים כל אחת מהקשות, שכן הזמן שאנו משקיע בכל שכבה הינה $O(1)$ זמן. אם כן, זמן הריצה הוא מס' השכבות, אם $1 + \ell$ הוא מס' השכבות זמן הריצה היה $O(\ell)$.

از מה קורה בתחילת האלגוריתם? בשלב הראשון של הפaza, בינויו את גוף השכבות L שיעלה $O(|E|)$ זמן.icut, נחפש מסלול קצר ביותר מס- t ב- L . אנו יודעים כי מסלול קצר ביותר מס- t ב- L הוא מסלול קצר גם ב- G_f . זה יעלה $O(|V|)$ כי אורך המסלול הוא לכל היותר $|V|$. המסלול קצר ביותר מס- t להמשיך להיפחס במסלולים קצרים ביותר ב- L . אך - ישנה בעיה. הבעיה היא שלאחר שמצאנו את המסלול הראשון, חלק מהקשות נהיות רווית וצריכות להמחק מגוף השכבות L . יתכן גם שצריכים למוחק קודקודים מסוימים מהגרף. ומה זה חשוב לנו? אמרנו שאנו בוחרים כלים לבחור קשת שרירותית בעת שמצאנו מסלול קצר ביותר - למה אמרנו שניתן לבחור שרירותית? כי לא משנה אם קשת נבחרת, תמיד בצד השני יהיה קודקוד שמייעים אליו ומשם ממשיכים, אך אם מהקנו קשת באטרציה הקודמת יתכן (מואוד) שהגישה של לקחת שרירותית לא תעוזר לנו ואנו נתקע - כי הקשת

השרירותית אליה הלאנו, היא קשת שמננה אין להתקדם (היה באיטרציה הקודמת דרך להתקדם, אך מחקנו את הקשת).
לכן, עלינו לפתח מנגנון שיבטיח שגם לאחר מציאת מסלול, גրף השכבות יהיה גרא מעודכן שעודנו גרא שכבות (ואז כן נוכל לחתך קשת שרירותית).

7.4 עדכון גרא השכבות

נסתכל על קשת $v \rightarrow u$ שצרכיה להמחק מהגרף. מה יכול לקרות?
מבחן u , יתכן ששונה קשת נוספת שיצאת מ- u , למשל (x, u) - אז המבחן של (v, u) לא משפיע על u כי באיטרציה הבאה ישן דרכים אחרים להתקדם משל דרכו x . אך, מה אם הקשת היחידה שיצאת מ- v היא (v, u) ? אם מוחק אותהCut - זה אומר שאין u אך להתקדם אל t בהמשך כי אולי יש הרבה קשותות שוכנות אליו אך אין קשותות שיצאות ממנו. הקשיי בה הוא שבשלב לפני u , אם נבחר בקש את u שרירותית אנחנו נתקע כי v אין אכן אך להתקדם.
במצב זה, אם $deg_{out}(v) = 0$, נרצה למוחק את u ולמחוק את כל הקשותות (w, u) שנכנסות אל u . ומה, אם היה קודקוד x שיש לו קשת (x, u) , וicut מוחקנו את u , והדרך היחידה יצאת מ- x היינה דרך x ,icut מוחקנו את u כי היא נכנסת אל u ומוחקנו את u - אזicut אנחנו צריכים למוחק גם את x והקשותות שיצאות ממנו; ומכאן שזה יכול להיות תהליך ארוך מאוד, כל המחיקה הוא ברקורסיה שנפרטת מכל המボאים הסטומים.
מה באשר ל- $s \rightarrow v$ שמוחקנו? יתכןCut שהקשת היחידה שנכנסת אל v הייתה (v, u) , במצב זה יתקיים כי $deg_{in}(v) = 0$, ולפי הגדרת גרא השכבות נctruck למוחק את v ואת צלעותיו.
נגיד רזאת פורמלי:

הגדרה:

1. מבוי סתום הוא מצב של קודקוד $t \neq v$ כך ש- $deg_{out}(v) = 0$.
2. מבוי סתום הוא מצב של קודקוד $s \neq v$ כך ש- $deg_{in}(v) = 0$.

עדכון של L בעקבות מחיקה של קשת: כל עוד קיים מבוי סתום כלשהו, מוחק אותו ואת כל קשותותיו.

כמה זמן לוקח הטיפול במובי סתו?

הבחנה: כל קודקוד נחה מבוי סתום פעם אחת בדיקות כל היותר בפואה. כיון, שברגע שנוחק אותו הוא לא יחזור להיות מבוי סתום. בנוסף, כל קשת נמחקת לכל היותר פעם אחת.
מכאן שסה"כ ישנים $O(|V|)$ קודקודים שנוחקו ו- $O(|E|)$ קשותות שנמחקו. אם כן, ההחלטה כיצד לוחק היא לוקאלית - אין צורך בסריקה נוספת של הגרא בעות שמצאנו מבוי סתום, אנחנו מוחקים את השכנים ומישרוב אליו, אין לנו סיבה לסרוק את כל הגרא לחפש את הבעה הבעיה לוקאלית.
לכן, סה"כ עלות כל העדכנים של L בפואה אחת עוללה $O(|E| + |V|) = O(|E|)$ זמן.

7.5 האלגוריתם של Dinic

להלן האלגוריתם של דיניק:

$\text{DINIC}(G = (V, E), s, t, c)$

- 1 initialize $f(u, v) = 0$ for all $u, v \in V$
- 2 **while** there exists a path from s to t in G_f
- 3 build layer graph L
- 4 **while** there exists a path P from s to t in L
- 5 augment f on P
- 6 update L by continuously removing dead-ends

הסביר על האלגוריתם:

בדומה לשיטה של פורד-פרקלטון, מתחילה את הזרימה להיות 0 . $f = L$. ההבדל בין האלגוריתמים של פורד-פרקלטון ושל DINIC יהיה במציאת מסלול השיפור שכן תיה מאוד מסויימת. לאחר מכן, נכנסים לילולת *while* כל עוד ישנו מסלול מ- s ל- t ב- G_f (בדיקה באמצעות BFS), בדומה לתנאי של EK .

בכל פזזה של האלגוריתם:

- A. בונים גרף שכבות L
- B. מתחילה איטרציה: כל עוד קיים מסלול P מ- s אל t ב- L :

 1. אם מצאנו - משפרים את f על המסלול. (זהה לתהליך ש庫ורה אצל פורד פרקלטון). מבצעים את הפעולות הבאות -

$$c_f(P) \leftarrow \min_{(u,v) \in P} \{c_f(u, v)\}$$

for each edge $(u, v) \in P$

$$f(u, v) \leftarrow f(u, v) + c_f(P)$$

$$c_f(u, v) \leftarrow c_f(u, v) - c_f(P)$$

$$f(v, u) \leftarrow f(v, u) - c_f(P)$$

$$c_f(v, u) \leftarrow c_f(v, u) + c_f(P)$$

update E_f

2. לאחר שSHIPRNU - מעודכנים את L כמו שאמרנו קודם: מורים את המボאים הסטומים.

נכונות האלגוריתם - נובעת ישרות פורד פרקלטון. נשים לב שלפי אדרטנס קארפ יהו לנו לכל היותר $1 |V| - 1$ גראפי שכבות.

מה באשר לזמן הריצה?

האתחול עולה $O(|V|^2)$. ישנו $O(|V|)$ פזזה ולכן שלב ב' יתבצע $O(|V|)$ פעמים. כל שלב

שזה:

בנייה גרף שכבות עלותו $O(|E|)$

- לאחר מכון נכנסים לולאות *while* של איטרציות. כל איטרציה עולה בבדיקה האם קיים מסלול $b(|V| + O(|E|))$ שיפור על המסלול ב(ℓ) זמן, וכן עדכון גרא השכבות עליוטו ($O(|E|)$ על כל הפהזה (!)).
- לא כל איטרציה.
- כמה איטרציות ישן בכל פאהזה? נניח שם' k זה הוא k . אם ישן k איטרציות בפהזה - אז כל פאהזה תעלה $\ell = O(|V| + k \times \ell)$ באשר ℓ הוא מס' השכבות כאשר $O(|V|)$. כמו כן, ישים $O(|V|^2 |E|)$ פאהזה, ונקבל כי זמן הריצה הוא:

$$|V| \times (|E| + k \times \ell) \leq |V| \times (|E| + |E||V|) = O(|V|^2 |E|)$$

כיוון ש $1 \leq |V| \leq |E|$ וכן $k \leq \ell$ (שכן ברגע שקשת נהיית רוויה בפהזה מסוימת, היא לעולם לא תהיה חלק ממסלול שיפור נוסף. אך בהכרח ישם לכל היותר $|E|$ מסלולי שיפור בכל פאהזה - איטרציות).

7.6 האלגוריתם של Hopcroft – Karp

קלט: רשת זרימה עם פונקציית קיבול על הצמתים. נגדיר פונקציה $b : V \setminus \{s, t\} \rightarrow \mathbb{R}^+$, ונגדיר לכל צומת u את הקיבול על הצומת. ככלומר, נניח שיש לנו קודקוד s עם קיבול 7, וכן נוכחות אליו קשתות עם קיבול 6 ו-10 בהתאם, וכן להזרים למשל 3 מתוך 6 ו-10 מותוך 10 וכן אכן הורם 7.

פלט: כיצד נפתחו זרימה מקסימלית באשר ישנו קיבול על הקודקודים?

מה שנעשה יהיה להגדיר גרף חדש:

$$G' = (V', E')$$

$$V' = \{S_{out}, S_{in}\} \cup \{u_{in}, u_{out} | u \in V\}$$

באשר, כל צומת תפוץל ל-2 צמתים: u_{in} ו- u_{out} , באשר הקיבול על הקשת $u_{in} \rightarrow u_{out}$ יוגדר להיות $b(u)$.Cut, כל מה שנכנס אל u ככלומר בגין החדש אל u_{in} , יוכל לעבור דרך הקשת $u_{in} \rightarrow u_{out}$ עם אילוץ הקיבול המתאים. כמו כן, נגדיר:

$$C'(u_{in}, u_{out}) = b(u), C'(u_{out}, v_{in}) = c(u, v)$$

מתקיים,

$$|V'| = 2|V| - 2$$

$$|E'| = |E| + |V| - 2$$

הסבר: נראה כי $2 - |V'| = 2|V|$ כי הכפלנו לכל צומת את הצומת עם צומת תאום, חוץ משני הצמתים s, t . וכן: $|E'| = |E| + |V| - 2$ שכן הקשתות בגרף המקורי יודם קיימות, וכן נוסףו $-|V|$ קשתות - קשת לכל קודקוד בגרף המקורי, פרט לשני הקודקודיים s, t .

מסקנה: בגרף החדש G' מתקיים $O(V) = O(E') = O(E)$ וכן

באופן ישיר מסקנה זו, נראה כי אם נרץ את דינ'יז' על גראף G' , שבעת הוא גראף עם קיבולות על הקשתות בלבד, נקבל אמונ ריצה על גראף זה של $O(|V|^2|E|)$. זה מtabסס על הטענה הבאה:

טענה: זרימה מקסימלית בגרף G' היא זרימה מקסימלית בגרף G עם הקיבולות על הצמתים.

תוספת: אם כן, אם הקיבולים שלמים ו-1, אז נוכל למצוא אלגוריתם טוב יותר. אותו נרצה לפרט עתה.

лемה 17. אם אוורך המק"ב ב- G מז אל t הוא x , אז חסם על הזרימה הגדולה ביותר הוא לכל $\frac{|V|-2}{x-1}$.
הוכחה שנייה פורמלית (בכיתה לא ניתנה הוכחה, הוכחה של). מהי הזרימה שלנו כעת? אם לכל קודקוד ישנו קיבול $b(u) = b, b \in \{s, t\}$, אז כל זרימה היא אוסף של מסלולים זרים בצמתים. ומכאן, ככל קודקוד יכול להשתתף במסלול אחד בלבד. כמה קודקודיים שורף כל מסלול? מסלול באורך x עבר דרך $x - 1$ קשתות, כך: $t \rightarrow u_{x-1} \times \dots \times u_1 \rightarrow s$, אלו $x - 1$ קודקודיים פנימיים, וכך כל מסלול שורף $x - 1$ קודקודיים שלא נוכל להשתמש בהם במסלולים אחרים. ישנו $2 - |V|$ קודקודיים פרט ל- s, t ולכן אם יש לנו t מסלולים, כל מסלול שורף $1 - x$ קודקודיים והמסלולים זרים אז מתקיים

$$t \times (x - 1) = |V| - 2$$

ובמילים אחרות,

$$t = \frac{|V| - 2}{x - 1}$$

מכאן, שערכן הזרימה הוא בדיקת מס' מסלולים זה, שכן בכל מסלול זורם ערך של 1. וכך.

נשים לב כי דינ'יז' מקרה פרטי של פורד פרקליטון וכן החסם של $O(|f^*||E|)$ תקף. אם כן, $|V| \leq |f^*|$ כיון שברתשית זו הזרימה יכולה להיות לכל היותר $|V|$, ומכאן לפי דינ'יז' הממש את אדרטנס קארפ זמן הריצה הינו $O(|E||V|)$. אם כן - נרצה לשפר.

מה קורה בפazaה של דינ'יז'? בעת שמצאנו מסלול מספר, נמחקה קשת אחת. כאן, נוחקות יותר קשתות. כל מסלול שנבחר יהיה מהצורה:

$$s \rightarrow \dots \rightarrow \dots \rightarrow \dots \rightarrow t$$

כל קשת בקיובל 1 בין (u_{in}, u_{out}) תמוחק לאחר הזרימה באיטרציה. ולכן - כל המסלול עצמו נמוחק באיטרציה זו (!). דרגה יוצאת של 0 $= u_{in}$ ודרגה כניסה של 0 $= u_{out}$ ולכן הצמתים

נמתקים וגם הקשתות שנוגעות בהם ומכאן שכל המסלול ימחק. ככלומר: בדינ'יז' רגיל, סך הזמן שלוקה בפאהה הינו $O(|V||E|)$ שכן בדינ'יז' רגיל אנו מארימים דרך מסלול, רק הקשת עם הקיבול הכי קטן נשרפת לגמרי ושאר הקשתות נשארות עם קיבול שיורי. הן יכולות להשתתף בהרבה מסלולים לאורך הפאהה. לכן פאהה אחת עולה שם $O(|E|)$. אם כן, אכן כל הקשתות במסלול נשרפות לגמרי - וכן כל המסלול עצמו נמתק. המשמעות היא שככל קשת יכולה להשתתף במסלול אחד בבדיקה בפאהה, וכן כל קשת נבדקת פעם אחת בבדיקה בפאהה - מה שעולה לפאהה $O(|E|)$. אם כן, מס' הפאות הינו $O(|V|)$, וכן סה"כ זמן הריצה יהיה $O(|V||E|)$. כאמור - לא שיפרנו כלום, הרי: האלגוריתם של פורד פרקליטון עובד ב- $O(|E||V|)$ כפי שהסבירנו. אז מדוע התעכברנו.

הערה. חשוב מאד לשים לב - בغالל הגדרת הגראף, המסלולים הינם זרים. וכך מחייבת קשת משפיעת על מסלול כלו, וכך בבדיקה קשת תהיה פעם אחת בפאהה ואז לא תהיה שוב. בנגדוד לדינ'יז' שם היא יכולה להופיע שוב.

הרעיון שיעבוד

נרצה להריץ את דינ'יז' רק על חלק מהפאות. נסמן את מס' הפאות הראשונות שנריץ כ- P . אם כן, זמן הריצה יהיה $O(P \times |E|)$ חלק זה של האלגוריתם. מה קורה לאחר P פאות? אורך המק"ב מ- s אל t ב- G_f , הוא לפחות P . מדוע? בכל פאהה, המסלול הקצר ביותר גדיל (לפי מה (11), וכן אורך המק"ב יהיה לפחות p . אם כן, כמה זרימה נותרה לנו להזרים? לפי מה (17), אם אורך המק"ב הכי גדול הוא x אי נותרו להזרים לכל היוטר $O(\frac{|V|}{p})$ ייחידות זרימה. ככלומר, ב- G_f , בicut מתקיים $\frac{|V|}{p} \leq |f^*|$. אם כן - מה שלא גרייך בicut את פורד פרקליטון? ראיינו כי זמן הריצה שלו יהיה $O(\frac{|V||E|}{p})$, וכך אצלו חלק זה יעלה $O(\frac{|V||E|}{p})$.

אם כן, זהו האלגוריתם. מה זמן הריצה שלו? ובכן - זה תלוי ב- P עצמו! נגידר את זמן הריצה כפונקציה של P , כדקלמן:

$$T(P) = O(FK) + O(Dinic) = P \times |E| + \frac{|V| \times |E|}{P}$$

אם כן, נרצה למצוא את זמן הריצה המינימלי, ככלומר את P עבשו ($T(P)$ מקבלת ערך מינימלי, ולכן, נגזרו:

$$T'(P) = |E| - \frac{|V| \times |E|}{P^2} = 0$$

$$P^2 = |V| \implies P = \sqrt{|V|}$$

זהו אכן ערך מינימלי. אם כן, נציג את האלגוריתם הבא:

- :Hopcroft – Karp($G = (V, E), b, c, s, t$)
א. נגידר את הגראף החדש G' כפי שתואר לעיל.
ב. נריץ את דינ'יז' על $\sqrt{|V|}$ פאות ראשונות.
ג. נריץ את פורד פרקליטון על שאר הפאות.

זמן הריצה: $O(\sqrt{|V|} \times |E|)$

שימוש טוב. כפי שנראה בדוגמה מטה, ניתן לבצע רדוקציה מזורית מקסימום לאיוג מקסימום בגרף דו צדי, ע"י הוספת קיבולות של 1 לצמתים.

7.7 זיוג מקסימום בגרף דו צדי

הגדה: יהיו $G = (V, E)$ גרף לא מכוון. תת קבוצה $M \subseteq E$ היא זיוג (או התאמה) של G אם $\deg_M(v) \leq 1$ לכל $v \in V$ כאשר $\deg_M(v)$ מסמל את הדרגה של v בגרף המושה ע"י M : $G' = (V, M)$. קודקוד v שדרגתנו $\deg_M(v) = 1$ נקרא קודקוד מזוג, ואילו קודקוד v שדרגתנו $\deg_M(v) = 0$ נקרא קודקוד לא מזוג.

הגדה: נאמר כי M זיוג מקסימלי - אם לא ניתן להוסף לו קשיות. כלומר, לא קיים $M' \subset M$ כך $|M'| > |M|$.

אלגוריתם פשוט למציאת זיוג מקסימלי: מתחילה עם הזיוג הריק, עבורים על כל קשת $E(u, v) \in E$ אם שני הצמתים פנויים מוסיפים אותה, אחרת לא מוסיפים. אלגוריתם חமדי ופשוט מאוד שירוץ בזמן לינארי.

בעיית זיוג מקסימום היא בעיית NP . לכן, נתמקד בבעיות זיוג מקסימום בגרף דו צדי. נרצה לתאר רדוקציה מבעיית זיוג מקסימום בעיית זרימת מקסימום בוגר.

יהי $G = (L, R, E)$ גרף לא מכוון דו צדי. נבנה רשת זרימה $G' = (V', E')$ באופן הבא:

$$V' = V \cup \{s, t\}$$

$$E' = \{(s, u) | u \in L\} \cup \{(u, v) | (u, v) \in E, u \in L, v \in R\} \cup \{(v, t) | v \in R\}$$

כמו כן, נגדיר את הקיבולות E' כ $\forall(u, v) \in E' \quad C(u, v) = 1$ להיות $(u, v) \notin E$ ולכל $(u, v) \in E$ כ $C(u, v) = 0$.

лемה 8. אם M הוא זיוג ב- G אז קיימות ב- G' זרימה f בעתה ערכים שלמים שערכה $|f| = |M|$ ומצד שני, אם f היא זרימה בערכים שלמים ב- G' אז קיימת M כ $|f| = |M|$ ב- G .

הוכחה: נוכיח צד ראשון. נניח כי M זיוג ב- G . נסתכל על הזרימה בה מכונים את כל הקשיות ב- M משמאל לימין, בנסוף לכל קודקוד $L \in u$ שמזוג ע"פ M נשים 1 בזרימה על הקשת (u, s) ובאופן דומה לכל קודקודים מזוגים $R \in v$ נשים 1 בזרימה על הקשת (v, t) . בזרירה זו הזרימה תשתמש בכל הקשיות המתאימות ל- M ובקהם. הזרימה נתו דרך החתך $\{s \cup L, \{t\} \cup R\}$ היא בדיקת $|M|$ כי כל הקשיות 1 ולכן $|f| = |M|$.

בכיוון ההיפוך, נסתכל על זרימה בשלמים f המוגדרת על רשת G' ונגידו:

$$M = \{(u, v) | f(u, v) > 0, u \in L, v \in R\}$$

נשים לב כי לכל $u \in L$ נכנסת רק קשת אחת ב- G' וקיבולה 1 ולכן כיוון שהזרימה בשלמים מובטח שהזרימה שנכנסת אל u היא אפס או אחד. משימור הזרימה, אנו יודעים כי קיבולות הזרימה הנכנסת

אל s היא 1 אם ומ"מ קיבול הזרימה היוצאת מ- s היא 1 וכיון שכל קשת היוצאת מ- s מגיעה לפחותו ב- R , אנו יודעים שהזרימה העוברת דרך s היא 1 אם ומ"מ קיים $v \in R$ כך $f(u, v) = 1$. לכן הדרגה של s ב- M היא לכל היותר 1. באופן סימטרי הדרגה של כל $v \in R$ היא אחד לפחות, לכן זו אכן זוג. מכאן,

$$|M| = f(L, R) = f(L, V') - f(L, L) - f(L, s) - f(L, t) = 0 - 0 + f(s, L) - 0 = f(s, L) =$$

$$f(s, V') - f(s, R) - f(s, t) = f(s, V') - 0 - 0 = f(s, V') = |f|$$

הлемה הקודמת טענה על שיקולות בין זוג לזרימות בערכים שלמים. כעת נטען, כי לרשות זרימה עם קיבולות בשלמים קיימת זרימת מקסימום בה הזרימה העוברת על כל קשת היא ערך שלם.
למה. אם לכל $(u, v) \in E$ מתקיים $c(u, v) \in \mathbb{N}$ או קיימת זרימת מקסימום בה לכל $(u, v) \in V \times V$ מתקיים $f(u, v) \in \mathbb{Z}$.
הוכחה: זרימת המקסימום שנמצאת ע"י שיטת פורד פרקלסן היא ציאת וזאת באינדוקציה על האיטרציות, והבנהה של מסלול שיפור הוא בערכים שלמים ושבוכם של ערכים שלמים הוא ערך שלם בשל סגירות לחבר וחיסור.

לכן, כיוון שיש התאמאה בין זוגים לזרימות, בפרט זרימת מקסימום ב- G' מותאמת לזוג מקסימום ב- G ולהפוך. מכאן, נוכל למצוא זוג מקסימום ב- G ע"י מציאת זרימת מקסימום ב- G' .

זמן הריצה: לפי שיטת פורד פרקלסן זמן הריצה הינו $O(|f^*||E|)$, כיוון שאצלנו $|f^*| = |M| \leq |L|$ זמן הריצה הינו $O(|E||V|)$.

7.8 זרימה אי זוגית

בזרימה הרבה מהשאלות הן תאורטיות עם הוכחות. נראה כעת דוגמה לתרגיל כזה:
תרגיל. תהי $G = (V, E)$ רשת זרימה עם פונקציית קיבול $c : V \times V \rightarrow \mathbb{Z}^+ \cup \{0\}$ וקידוד מקור s וקידוד עד $t \in V$. בנוסף עבור הקשת $e = (x, y) \in E$ מתקיים כי הקיבולות שלה היא ערך אי זוגי. לכל קשת אחרת, $e' \neq e \in E$ מתקיים שהקבולות שלה היא ערך זוגי. תהי f^* זרימת מקסימום ברשת. הוכח או הפרך: $|f^*|$ בuttle ערך אי זוגי \iff הקשת e רויה.

נבחן, כי לפי $MFMC$ בזרימת מקסימום כל הקשתות בחנת (s, t) הן רוויות. נוכיח מסקנה זו. לפי משפט זה קיימים חנת (s, t) נסמיין (S, T) כך $|f^*| = c(S, T)$. לפי ההרכאה מתקיים כי $c(S, T) = f(S, T) = f(S, T) - |f^*|$ ולכן $c(S, T) = f(S, T)$.

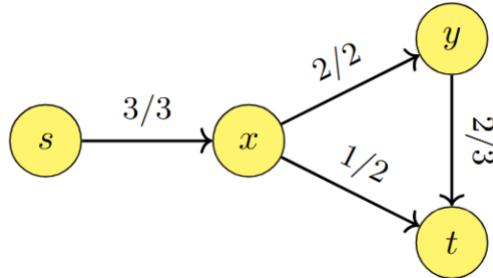
$$f(S, T) = \sum_{x \in S} \sum_{y \in T} f(x, y) \leq \sum_{x \in S} \sum_{y \in T} c(x, y) = c(S, T)$$

כיון שיש שוויון בהכרח לכל $T \times T$ מתקיים $(x, y) \in S$ מתקיים $c(x, y) = f(x, y)$. כלומר, כל קשת שחוצה את החנת היא רויה. נניח בשיליה כי הקשת $(x, y) < c(x, y)$. אם כן, ניתן שקיימת לא רויה ממשמע כי הקשת אינה חוצה את החנת, שכן כל קשת שחוצה את החנת היא אכן רויה. ולכן, בהכרח הערכים של הקשתות שחוצות את החנת הוא זוגי. שכן, כל שאר ערכי הקיבולות זוגיים וכל קשת שחוצה את החנת רויה. אם כן, סכום הקשתות שחוצות את החנת הוא ערך הזרימה, אך נקבע

כי ערך הזרימה הוא סכום של זוגים ולכז זוג. אם כן, מכאן נקבל סטירה. לכן $e = (x, y)$ אכן רויה.

תרגיל נוסף. ללא הנ吐ן על זוגיות ואי זוגיות. נניח שקשת $(x, y) = e$ היא רויה. האם זה גורר שהיא בהכרח נמצאת בחתך (t, s) מינימלי ברטה? התשובה היא לא! הבהרה - זהו תנאי הכרחי אך לא מספיק. כלומר: אם קשת שייכת לחתך מינימום ברטה היא בהכרח רויה, אך אם היא רויה זה לא אומר כלום.

דוגמה נגד:



נבחן כי ברטה זו החתך המינימלי היחיד הינו $\{s\}, \{x, y, z\}$. סכום הקיבולות בחתך זה הוא 3 וזה גם ערך הזרימה המקסימלית. אם כן, מתקיים כי $f^*(x, y) = c(x, y) = 1$ - ככלומר היא רויה, אך היא לא חוזה אף חתך מינימלי, בסטירה לטענה.

8 הרצאה 8: BMM , מושלשים וקל בעד

BMM 8.1

בכפל מטריצות רגיל מתקיים לפי עדי בן צבי -

:Boolean Metrix Multipcation (BMM) שנkirא

$$C_{ij} = \bigvee_{k=1}^n a_{ik} \wedge b_{kj}$$

כלומר, כאשר מייצגים מטריצות בוליאניות אנחנו מסתכלים על המיקום h_{ij} . הוא מכפלה של השורה ה i במטריצה A והעמודה ה j במטריצה B , מספיק שיהיה קיים אינדקס אחד k עבורו הערך $a_{ik} \wedge b_{kj}$ בשורה ה i והערך h_k בעמודה ה j הוא 1, אי גדייר, 0 אחרת, $c_{ij} = 1$.

נסמן בא את האקספוננט של האלגוריתם הכי מהיר שקיים לפתורן בעיית כפל המטריצות. בהכרח, $n \leq 2.37287$ כי כו"ם ידוע אלגוריתם שפותר בזמן זה, וכן נקבל סה"כ כי:

$$2 \leq \omega \leq 2.371339$$

מכאן, האלגוריתם הכי מהיר לכפל מטריצות ריבועיות מגודל $n \times n$ עולה $O(n^\omega)$ זמן.

8.1.1 BMM קומבינטורי והשערת BMM

המושג של אלגוריתם קומבינטורי אינו מוגדר היטב. עם זאת, ניתן לומר מהו אלגוריתם לא קומבינטורי. האלגוריתמים של FMM משתמשים בפעולות חישור והמקדמים הקבועים בזמן הריצה בדרך כלל גדולים מאוד (!). מכאן - שבדרכּ כל זה לא מעשי להשתמש באלגוריתמים האלה (פרט לאלגוריתם של שטרנסן).

מצד שני, אלגוריתמים "קומבינטוריים" נוטים להיות מהירים מבחינה מקדמי הקבועים וגם הם יחסית אלגוריתמים פשוטים. מדוע זה טוב? קל למשש אותן. ברור לנו כי אלגוריתם קומבינטורי הוא אלגוריתם שלא ישמש בחסרות, ככלומר בחישור: רוצים להשתמש בכלים יותר "שיררים" - החסורה היא פוליה שمبטלת משהו שביצענו יותר מדי.

הנאיבי שעליו $O(n^3)$ מאוד קל למימוש, וכן המהיר ביותר (הקומבינטורי) שקיים היום בסיבוכיות $O(\frac{n^3}{\log^4 n})$.
נרצה להתמקד ב- BMM שואלי (!!?) יותר קלה מה FMM אך בוודאי לא יותר קשה.

אחת מהשאלות הכני מעניינות ופתוחות כיום בעולם הקומבינטורי ומדעי המחשב היא - **האם קיים אלגוריתם "קומבינטוררי" ל- BMM שזמן הריצה שלו הינו $O(n^{3-\varepsilon})$ כאשר $\varepsilon > 0$?**

נראה כי אפילו אם ימצאו אלגוריתם שרצ' בסיבוכיות זמן $O(n^{2.9999\dots})$ בהכרח יתקיים

$$n^{2.9999\dots} \ll \frac{n^3}{\log^4 n}$$

אנחנו נתעלם מפקטורים של $(\log n)$ וכו', נתעניין רק באקספוננט. מכאן נגידר סימון חדש () שיציין התעלומות זאת. לדוגמה:

$$O(n^2 \log n) = \tilde{O}(n^2)$$

$$O(\frac{n^3}{\log^4 n}) = \tilde{O}(n^3)$$

בשנת 2024 הוכיחו כי קיים אלגוריתם קומבינטורי שזמן הריצה שלו הוא $O(\frac{n^3}{2\sqrt[3]{\log n}})$.

השערת BMM : לכל קבוע $0 < \varepsilon$ לא קיים אלגוריתם קומבינטורי שזמן הריצה שלו הוא $O(n^{3-\varepsilon})$.
נבחן, כי דרך אחרת לכתוב את ההשערה היא שככל אלגוריתם קומבינטורי ל- BMM דרוש זמן $(1+o(1))n^{3-o(1)}$ קטן מחייב את הפקטורים של $(\log n)$, נבחן כי $n^{3-o(1)} \notin n^{2.99999999}$ בעותת ההשערה זו, ניתן להוכיח חסמים תחרתניים לאלגוריתמים "קומבינטוריים" עבור כל מני בעיות (כתלות בנסיבות ההשערה).

מי השערה? השערה יוצאה לאחר שניסו הרבה זמן לפתור בעיה ולא הצליחו. האם בהינתן השערה מסוימת, זה אומר שהיא נכונה? לא. אם היה הוכחה לשערה - היא לא הייתה השערה. ולכן, אם בהמשך יוכלו אותנו הם כבר לא יהיו השערה. למשל, חוקי ניוטון הם רק השערה (אין הוכחה). אם כן: **כל הנואת צריך להשתמש בשיטות אלגוריתמיות חדשות** (שיתכן שעוד

לא המצאננו) על מנת להפריך את ההשערה. זה פותח עבורנו חקר של הסתכלות על משפחות מיוחדות מיחדות של קלטימס - זה רלוונטי עבור פרקטיקה: אם למשל בכפל מטריצות נתון כי $A = \{a_{ij} = 1\}$ (כל המטריצה אוחדות) אז הבעה הייתה הופכת לקללה הרבה יותר - ב- $O(n^2)$. זו משפחה מיוחדת של קלטימס.

על סמך השערות ניתן להוכיח חסמים תחתונים מותנים. ובאנגלית: *conditional lower bounds*: למשל, תחת ההנחה $P \neq NP$ אפשר להוכיח כי בעיות רבות אין ניתנות לפתרון בזמן פולינומיyal: אלא בהכרח בזמן אקספוננציאלי.

במדעי המחשב אנחנו דיברים בהוכחה חסמים תחתונים לביעות. אם כן, הזמן היחיד שכן הצלחנו להוכיח הוא חסם תחתון למספר מובסש השוואות. אך, בהינתן השערה מסוימת נוכל כן להניא חסמים תחתוניים כלשהם.

8.2 זיהוי משולשים בגרף

הגדרה: יהי $G = (V, E)$ גרף. מסלול (v_0, v_1, v_2, v_0) יקרא משולש. כמובן, משולש הוא מעגל בגודל 3.

קליט: $G = (V, E)$ גראף לא מכון.

פלט: ישנו מס' אפשרויות.

1. האם G יש משולש?

2. אם G יש משולש, מצא אחד שכזה.

3. דוח על כל המשולשים שיש ב- G .

אנחנו נתמקד בבעיה 1: האם G יש משולש? נניח כי G מוצגת ע"י מטריצת שכניות M . ניתן לבדוק אם G יש משולש ע"י בדיקה של M^3 בכפל BMM .

טענה: $\exists_{1 \leq i \leq n} M^3[i][i] = 1 \iff$ (באילסון) G יש משולש.
מסקנה: ניתן להזות משולש ע"י 2 חישובים של M , BMM , בדומה לא קומבינטורית נוכן לטעון שסיבוכיות האלגוריתם הינה $O(n^\omega)$. בדומה קומבינטורית, נטען כי קיימים אלגוריתם קומבינטורית בזמן $\tilde{O}(|V|^3)$.

שאלת: האם ניתן להזות משולש בגרף G בזמן יותר מהיר פולינומי?

טענה (משפט 2): אם קיימים אלגוריתם קומבינטוררי שפותר זיהוי משולשים בזמן $\tilde{O}(|V|^{3-\varepsilon})$ אז קיימים אלגוריתם קומבינטוררי שפותר את BMM בזמן $\tilde{O}(n^{3-\frac{\varepsilon}{3}})$.

מסקנה: על סמך השערת BMM הקומבינטורית, לא קיימים אלגוריתם קומבינטוררי שפותר זיהוי משולשים בזמן $\tilde{O}(|V|^{3-\varepsilon})$.

הוכחה:

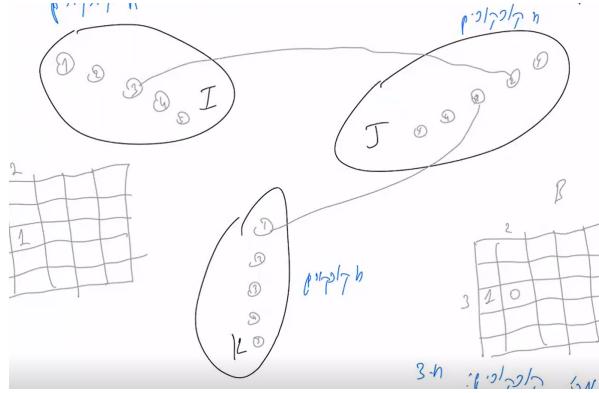
קליט: A, B מטריצות כוילאיות מוגדי $n \times n$ וכן אלגוריתם קומבינטוררי שפותר זיהוי משולשים בזמן $\tilde{O}(|V|^{3-\varepsilon})$

פלט: כאשר הכפל הוא $C = A \times B$

נכיה גראף תלת צדי (כמו זו צדזין, רק של שלושה חלקי), אשר החלקים יקראו K, I, J , וכל חלק מס' שווה של קוזקוזים ונוסיר קשת בין I ו- J לבין J ו- K ($v \in I$ וכיום $v \in J$ וכיום $v \in K$). אשר כפלי ה- K מתחיקיס בה

$A[v][u] = 1$, $B[u][v] = 1$, וכזכורה תראה קשת בין I ו- J וכיום $v \in K$ וכיום $v \in J$.

פורמלית מעט יותר: לכל $1 \leq i, j \leq n$ וטורי $A_{ij} = 1$ וטורי קשת מהקוזקוז ה- i לקוזקוז ה- j . כאמור, לכל $1 \leq i, j \leq n$ וטורי קשת מהקוזקוז ה- i לקוזקוז ה- j .



סיבלו גורף חדש, מתקיים בו $|E| = O(n^2)$ (שכו מס' הצלעות הוא כמס' ה-1 ב- A , B , במקורה הגורע המטריצות כלו' אחות). כלומר $n^2 \leq |E| \leq 3n^2$ צלעות כמו כן, נסיף את כל הקשות האפשריות בין K ל- I . מכאו קיבלו שבען I יש ממשי n^2 צלעות - ככל אחו' מהס יש n קוזקושים.

טענה - עבור $i \in I$ ו- $k \in K$ הששת (i, k) נמצאת כמשולש $\iff C[i][k] = 1$. (C היא מטריצה הפלט של המכפל הכלולאי).

נניח כי הקשת (i, k) נמצאת כמשולש. אז, קיים איזקס j עבורו קיימות הקשות (j, k) ו- (i, j) . מכאו, לפי ההגדרה $B[k, j] = B[j, k] = 1$ ו- $A[i, j] = A[j, i] = 1$.

$$C_{ik} = \bigvee_{m=1}^n a_{im} \wedge b_{mk}$$

בפרט עבור $j = m$ נסתכל ונוכיח: $a_{ij} \wedge b_{jk} = 1 \wedge 1 = 1$ ונקבל:

$$C_{ik} = \left(\bigvee_{m=1, m \neq j}^n a_{im} \wedge b_{mk} \right) \wedge 1 = 1$$

ומכאן $C[i][k] = 1$. נניח כי $C[i][k] = 1$. אז, קיים איזקס j עבורו $a_{ij} = 1 \wedge b_{jk} = 1$. קיימת כי כל הקשות בין I ו- K קיימות, מכאו שקיימים שני הקשות האחרות נקבען כי (i, k) כמשולש.

נסיוו ראשון (לא יעכז):

- אתחל $C = 0$
- כל עוד G יש משולש $I \times J \times K$:
1. $c_{i,k} = 1$
2. הורד את הקשת (i, k) מהגרף.

הכחיה ראשונה: מס' האיטרציות הוא $O(n^2)$, שכו כל איטרציה מורידה קשת אחת בין I ל- K - ולכו אחריו $O(n^2)$ פעמים אין יותר משולשים. כלומר, זמן הריצה הוא n^2 כפול זמן הריצה של אלגוריתם ליזיה משולשים. מכאו נקבל:

$$\tilde{O}(|V|^{3-\varepsilon}) \times n^2 \implies_{|V|=3n} \tilde{O}(3^{3-\varepsilon} \times n^{3-\varepsilon+2}) = \tilde{O}(n^{5-\varepsilon})$$

זמנן הריצה לא תואם למזה שרצינו לעשות. חכל. בעהו נוספת: האלגוריתם שאחנו יסייעו לבנות מזא צמושש, האלגוריתם עליו זיכרנו כ"קופסה שחורה" יודע **לזהות משולש**.

כעת נרצה לשפר את זמנו הריצה. משתמש באותם I, J, K אך הפעם חילק כל אחד מהם ל t חלקים: $(I_1, \dots, I_t), (J_1, \dots, J_t), (K_1, \dots, K_t)$ כל עוד קיים משולש בגרף שמשורה עי' השלשה הנוכחות הקוווט:

- האלגוריתם:
- א. נאתחל $C = 0$
- ב. עכו כשלשה (I_x, J_y, K_z) נורץ את האלגוריתם הקוווט:
- 1. קבע $C_{ik} = 1$
- 2. נורץ את (i, k) מהגרף G .

ובעבירות: תחילת נורץ את (I_1, J_1, K_1) . אם נמצא משולש בין חלקים אלו אנחנו נורץ את הקשת (i, k) הרלוונטי. לאחר מכן נורץ את האלגוריתם שכעל אותה שלשה. אם נמצא עוד משולשים - נקבע תחילה דומה גם כן. לאחר מכן נורץ את (I_1, J_2, K_1) וכן הלאה - על כל שלשה.

נכונות: האלגוריתם עובד מאותה הסיבה שהקוזם עבד - מוצאים את כל המשולשים בגרפים המושרים, ופזרט כל הגופים המושרים מהווים יחד את הגרף כולו G .

זמן הריצה:
 כמה שלשות יש בגרף? יש t אפשרויות כ- k, j, i . כלומר שיש t^3 שלשות.
 נאמר שהריצה של אלגוריתם למספר משולש נסילה אם התשובה היא שוויי משולש, ואחרות היא הצלחה.
 כל כשלו מוביל לשלשה חדשה, וכל הצלחה קובעת ערך C . כמה פעמים ניתן להגעה לכשלו? לכל היותר t^3 כשלונות. כמה פעמים אפשר להצלחה? n^2 לכל היותר כטמי התאים ב- C וכן לא ניתן שתהיה הצלחה פעמיים על אותו תא, כי אנחנו מורזיזים את הקשת (i, k) . כמו כן, ככל אחת מ- t^3 השלשות ישנים $\frac{n}{t}$ קוזזים עליהם מופעל האלגוריתם.
 מס' הפעימות שהאלגוריתם מירץ את האלגוריתם יזרום משולשים היו $n^2 + t^3$. כמו כן, את האלגוריתם הכליל הוא מפעיל בכל פעם על גוף מגודל $\frac{n}{t}^3$, כלומר שזמנו הריצה יהיה

$$(t^3 + n^2) \times \tilde{O}\left(\left(3\frac{n}{t}\right)^{3-\varepsilon}\right) =$$

$$3^{3-\varepsilon} \left(t^3 \left(\frac{n}{t}\right)^{3-\varepsilon} + n^2 \left(\frac{n}{t}\right)^{3-\varepsilon}\right) =$$

אם נשווה את שווי הביטויים נקבל כי

$$t^3 \left(\frac{n}{t}\right)^{3-\varepsilon} = n^2 \left(\frac{n}{t}\right)^{3-\varepsilon}$$

$$t^3 = n^2 \implies t = n^{\frac{2}{3}}$$

עכור $n^{\frac{2}{3}}$ $t = n$ נקצל את זמן הריצה:

$$n^2 \times n^{\frac{3-\varepsilon}{3}} + (n^{\frac{2}{3}})^3 n^{\frac{3-\varepsilon}{3}} = \tilde{O}(n^{3-\frac{\varepsilon}{3}})$$

לסיום, יצירנו אלגוריתם שיזע להחזיר את מטירעת המכפל $\tilde{O}(n^{3-\frac{\varepsilon}{3}})$ כאשר השתמשנו באלגוריתם שפיצה משולש.

הערה נוספת. נבחין כי באמצעות (והאינטואיטיבית) לחלוקת t היא להקטין את גודל הקלט של הגרף שMRIיצים עליו את האלגוריתם למציאות משולשים בכל שלב - וכן רצינו להקטין את גודל הגרף זהה בכל שלב. אם כן, אנחנו בכל שלב בוחרים חלק $K \times J \times I$ (x, y, z) ובכל אחד מהם יש חלקים ולבן סה"כ t^3 .

נבחין, כי הנחנו שהאלגוריתם גם מזהה משולש וגם מחזיר אותו. מדוע? נתבונן בلمמה הבאה:

נבחין בעת, כי בסוף נגיע ל-4 חלקים בגודל 1 (תנאי העצרה) ונמצא את המשולש הספציפי. סה"כ נשים לב שבאותו זמן של האם קיים משולש ניתן למצוא אותו גם, כמובן.

8.3 (מציאת משולשים) בגרפים דילילים (כל כבד)

ראינו כי $(\omega|V|)O$ בגרף כללי. נראה כתת אלגוריתם כgraf עם מעט קשות, כתלות ב $|E|$. זה לא אלגוריתם קומבינטוריה אלא רגיל.

נדיר פרמטר בשם τ (בහמץ נקבע אותו ממש).

הגדרה: נאמר כי קודקוד u הוא קודקוד כבד אם $\tau \geq \deg(u)$. אחרת - נאמר כי u הוא קודקוד כל.

טענה: מס' הקודקודיים הכבדים הוא לכל היוטר $\frac{2|E|}{\tau}$.
הוכחה: נניח בשילhouette כי מס' הקודקודיים הכבדים הינו $< \frac{2|E|}{\tau}$ ונקבל כי סכום דרגותיהם הינו $< \tau \times 2|E| = \frac{2|E|}{\tau} \times 2|E|$ בסתייה למלה לחיצת הידיים.

נראה אלגוריתם שמאזן את שני המקרים - קודקודיים קלים וכבדים. כאמור, נראה אלגוריתם לקלים ולכבדים והאלגוריתם בסוף יאזור את שניהם.

עבור u קודקוד כל (וסה"כ עבור הקלים): נὔבור על כל זוגות השכנים $(u, v) \in E$: לכל היוטר $\deg(u)$ זוגות שכאלו. וכן, אם $(x, y) \in E$ נחיזר כן: ישנו משולש. (שכן, x שכן של u וגם y שכן של u וישנה קשרת (x, y) וסה"כ משולש (x, y, u) .).

$$O(\sum_u \deg^2(u))$$

עבור כל הכבדים יחד: יהיו \hat{G} הגרף שמושרחה ע"י כל הקודקודיים הכבדים. מתקיים $|\hat{V}| \leq \frac{2|E|}{\tau}$.
נשתמש FMM על \hat{G} (עליה בשילישית ונעזר בטענה שבאלכסון יש אחד אמ"מ משולש). נקבל זמן ריצה: $O((\frac{|E|}{\tau})^\omega)$.
סה"כ, נריץ את שני האלגוריתמים יחד. נבחין כי ישנו שני סוגי של משולשים: משולש שכל הקודקודיים בו כבדים. נגלה אותו בשלב של FMM : אחרת, יש לפחות קודקוד אחד קל במשולש ונגלה אותו בשלב השני שנריץ עבור הקלים.
סה"כ, זמן הריצה יהיה:

$$O\left(\sum_u \deg^2(u)\right) + O\left(\left(\frac{|E|}{\tau}\right)^\omega\right)$$

$$O\left(\sum_u \deg^2(u)\right) \leq O\left(\sum_u \tau \times \deg(u)\right) = \tau \sum_u \deg(u) = 2\tau|E|$$

ולכן זמן הריצה:

$$2\tau|E| + \left(\frac{|E|}{\tau}\right)^\omega$$

נשווה כעת את שני הביטויים:

$$\tau|E| = \left(\frac{|E|}{\tau}\right)^\omega \implies \tau^{\omega+1} = |E|^{\omega-1} \implies \tau = |E|^{\frac{\omega-1}{\omega+1}}$$

מכאן, שה" τ " זמן ריצה האלגוריתם:

$$\left(\frac{|E|}{\tau}\right)^\omega = \left(\frac{|E|^{\frac{\omega-1}{\omega+1}}}{|E|^{\frac{1}{\omega+1}}}\right)^\omega = \left(|E|^{1-\frac{\omega-1}{\omega+1}}\right)^\omega = O\left(|E|^{\frac{2\omega}{\omega+1}}\right)$$

הבהרה. ניתן לבצע את האלגוריתם בכל גרף, יתכן שהוא לא כדאי בגרפים שאינם דילילים. נבחן כי אם יום אחד יגלו כי אכן $\omega = 2$ אז האלגוריתם רץ בזמן ריצה $O(|E|^{\frac{4}{3}})$ - מדהים!

8.4 בעיית דיווח המשולשים (כל כבץ)

קודם לכן, דנו באלגוריתם ליהויי משולשים בגרף. כעת, נרצה לדון בדיווח על כל המשולשים בגרף. האלגוריתם שנדון בו יפעל בצורה מעניינת של חלוקת הקלט לחלקים קלים ו"כבדים" - ובמהשך נציגו שיטה זו (כל כבץ) לבעה אלגוריתמית נוספת.

הגדרת הבעיה: דיווח משולשים בגרף

קלט: גרף $G = (V, E)$ לא מכובן

פלט: כל שלושות הקזוקדים (u, v, w) כך $(u, v, w), (v, w), (w, u) \in E$

פתרון ראשון:

:*Algo1*($G = (V, E)$)

: $u, v, w \in V$

א. לכל $(u, v, w) \in E$ ב. בדוק אם $(u, v) \in E \wedge (v, w) \in E \wedge (w, u) \in E$ המצב דוחה על כמשולש.

כל לראות שהאלגוריתם מבצע את הנדרש, זמן הריצה שלו $O\left(\binom{|V|}{3}\right) = O(|V|^3)$

פתרון שני:

זמן הריצה בפתרון הראשון לא היה תלוי בקשנות הגרף. לכן, נרצה שהאלגוריתם יתחשב בקשנות הגרף. בכל מושולש - ניתן להסתכל עליו כקשת בסיס (v, u) וקודוקוד שלישי w שמחובר בשתי קשותות אחרות לקשת הבסיס. לכן נציג את האלגוריתם הבא:

```
:Algo2( $G = (V, E)$   
:  $w \in V$   
: א. עברו כל  $(u, v) \in E$   
: ב. עברו כל  $u, v, w \in E$  דוח על  $(v, w) \in E \wedge (u, w) \in E$ . אם .1
```

אכן האלגוריתם מבצע את הנדרש, וזמן הריצה כמובן $O(|V||E|)$.

פתרון שלישי:

במהשך לפתרון הקודם, נראה כי אם לקצחות הקשת u ו- v יש מעט שכנים, אין טעם לעבור על כל קודוקודי הגרף כקודוקודים אופציונליים לשגירת המשולש ומספיק לעבור למשל על השכנים של u ולבדוק אם הם גם שכנים של v ולהפוך. מבחינה מתמטית, קודוקודים w שמהווים משולש יחד עם u ו- v הם בדיקת הקודוקודים שהם שכנים גם של u וגם של v . כלומר, $(u \cap v) \in \Gamma(w)$. אם הגרף מייצג ע"י רשימות שכניות, וכל הרשימות ממיניות לפי סדר גלובלי של קודוקודי הגרף, אז ניתן לחשב חיתוך זהה במעבר אחד מסונכרן על שתי הרשימות שעלותו $O(\deg(v) + \deg(u))$. להלן אלגוריתם -

```
:Algo3( $G = (V, E)$   
:  $(v, u) \in E$  על כל הקשותות :  
1. חשב את החיתוך  $\Gamma(v) \cap \Gamma(u)$   
2. לכל  $(u, v) \in \Gamma(v)$  על המושולש  $(u, v, w)$ : דוח על  $w$ 
```

הכנות מיידית, וזמן הריצה עולהו $O(\sum_{(u,v) \in E} (\deg(u) + \deg(v)))$

חסם תחתון לזמן הריצה: במקרה הגרע, ישנו $O(\binom{|V|}{3})$ משולשים ובקרה זה זמן ריצה של כל אלגוריתם שיפטר את הבעיה לא יכול להיות פחות מאשר $O(|V|^3)$. מה אם אס c , חסם זה תקף c^m מסטכלים על מס' הקודוקודים כפרמטר יחיד לזמן הריצה. מה נחש את זמן הריצה של אלגוריתם לבעה כפונקציה של מס' הקשותות ומס' הקודוקודים? הדוגמה של גраф עם $\binom{|V|}{3}$ היא עם $\Theta(|V|^2)$ כלומר זמן הריצה יכול להיות מוגדר גם כ- $O(|E|^{1.5})$.

נרצה למצוא אלגוריתם שזמן הריצה שלו תלוי כמה שיותר במס' הקשותות וככמה שפחות במס' הקודוקודים.

זאת, כיוון שכן שכך נזווית יותר מגזר דليل עם מעט קשותות. אם משתמשים על הדוגמה של חיתון לגרף מלא מבחינת משולשים רואים כי זמן ריצה כפונקציה של $|E|$ חייב להיות $O(|E|^{1.5})$. נרצה זמן ריצה שייגיע לזמן שכזה על כל גראף. באלגוריתם החלשי קיבלנו זמן ריצה שתלוי בדרגת הקודוקודים, נשים לב שדרך אחרת לחושב על דרגות הקודוקודים היא להזכיר בכך שככל קשת תורמת 1 לדרגנה של שני קצוותיה, לכן אפשר לחושב שכאשר ידוע על גראף עם $|E|$ קשותות ישנו $2|E|$ אסימוניונים אותם יש לחלק בין $|V|$ הקודוקודים. נפרמל:

הגדרה. יהיו $G = (V, E)$ גראף לא מכוכן. קודוקוד $V \in u$ נקרא כל אם מתקיים $\deg(v) \leq \sqrt{|E|}$. ואחרת, $\deg(v) > \sqrt{|E|}$ וקודוקוד זה נקרא כבד.

למה. בכל גראף ישנו לכל היותר $2\sqrt{|E|}$ קודוקודים קבועים. הוכחה: מס' הקשותות בגראף הינו $|E|$. לכן מלעתה לחיצת הידיים סכום הדרגות בגראף הוא $2|E|$. נניח בשלילה כי ישנו יותר מ- $2\sqrt{|E|}$ קודוקודים קבועים. אז, סכום דרגותיהם $< 2\sqrt{|E|} \times \sqrt{|E|} = 2|E|$ בסתירה.

בעת, נחלק את המשולשים לשני סוגים:

- א. משולש שמכיל קודקודים בלבד
- ב. משולש שכל קודקודיים קלים.

חיה, נפעיל את האלגוריתם השני על הקודקודיים הקלים בלבד. כמובן, עברור על כל הזוגות של קודקוד כבד וקשת כלשהי ונבדוק אם הם יוצרים משולש. בשלב השני, עברור על כל הקשתות (v, u) כך שגמ u וגם v קלים, ונחתוך את רשימת השכניםות שלהם. (ושוב כמו באלו 3 אנו נניח כי רשימת השכניםות נתונה כרשימת שכנוויות ממוינית).

:Algo4($G = (V, E)$)

א. קבע לכל קודקוד $V \in V$ אם הוא קל או כבד.

ב. עברור על כל קודקוד כבד $w \in w$:

.1 אם $(u, w) \in E$ ו- $(v, w) \in E$ דוחה על המשולש w, v, u .

.2 עברור על כל הקשתות $(v, u) \in E$.

.1 חשב את החיתוך $\Gamma(v) \cap \Gamma(u)$.

.2 לכל $u \in \Gamma(v) \cap \Gamma(w)$ דוחה על המשולש w, v, u .

למה .4. $Algo4(G = (V, E))$ מדווח על כל המשולשים ב- G ורק עליהם.

הוכחה: ראשית נשים לב לכך שהאלגוריתם לא מדווח על שלשה שאינה משולש, כיון שהוא מורכב

מאלגוריתמים שכבר ראיינו שמחשבים משולש.

יב. יי (v, w, u) משולש. אם לפחות אחד מהקודקודיים כבדים, בה"כ w נדוחה על המשולש בסעיף

ב'. אחרת, כל הקודקודיים קלים, אז במעבר על (u, v) בסעיף ג' נמצאת $(u, w) \in \Gamma(v) \cap \Gamma(w)$ ונדוחה

בג' על (v, w, u) .

זמן הריצה: השלב הראשון עולה $O(|V|_{heavy} \times |E|) = O(2\sqrt{|E|} \times |E|) = O(|E|^{1.5})$, השלב

השני עולה

$$\sum_{(u, v) \in E} (\deg(u) + \deg(v)) \leq \sum_{(u, v)} 2\sqrt{E} \leq |E| \times 2\sqrt{E} = O(|E|^{1.5})$$

וסה"כ זמן הריצה הינו $O(|E|^{1.5})$.

8.5 מרכיב האמיניג עבור א"ב כללי (כל כביד)

נרצה לפטור את בעיית מרכיב האמיניג עבור א"ב כללי ככלומר מצב בו $(\Sigma) = O(n + m) = O(n + m)$. באופן דומה, נחשב את מס' ההתאמות עבור על מיקום מסו'Ai התאמתים כיון שנitin לעבור מההתאמות לאי ההתאמות ב- (n) . א. באופן הנאי, עבור האלגוריתם שראינו קודם לכן אם נרצה עבורו ($\Sigma = |\Sigma|$) נקבל זמן ריצה של $O(n^2 \log m)$.

לשיט הפשטות, נחשב על מקרה בו בתבנית יש מופיע אחד של a במיקום השלישי נניח.Cut, ניתן לספר את ההתאמות של a בין התבנית לטקסט באופן הבא: נבצע סריקה של הטקסט, ובכל פעם שנראה a במיקום $i + 1$ לモונה ההתאמות $[i - 3 + 1]$ לפניה האינדקס אותו אנו שורקים. אם יש שני תווים a במיקומים x ו- y , ניתן לבצע סריקה דומה כאשר כל פעם שראוים a במיקומים $+1$ במיקומים $[x - y + 1]$ ו- $M[i - x + 1]$. נשים לב, שאם התו b נמצא בносף במיקום אחד בחרוזת, ניתן באותה סריקה של הטקסט למנות את מופיעי a וגם מופיעי b כיון שכל פעם ששורקים אותם באותו טקסט צריך לבודד אותם. ציריך לבצע הוספות למונחים רק לפי המונחים של האות הספציפית הזאת בתבנית. ככלומר, מציין $M[k]$ כמו להתאמות יש במיקום c אשר מתחילה ב- c .

באופן פורמלי, נסמן לכל $\Sigma \in \sigma$ את מס' המופיעים של σ בתבנית $C(\sigma)$ ואת המופיעים עצמים כאינדקסים $i_{\sigma c(\sigma)}, i_{\sigma 2}, \dots, i_{\sigma}$. האלגוריתם בשלב ראשון יבנה מבנה נתונים המכיל את כל האינדקסים הללו לכל האותיות ובשלב השני יבצע מעבר אחד על הטקסט ובכל מקום j כאשר האות

המופיע היא σ כלשיי יוסיף לכל המוניים $j - i_{\sigma 1} + 1, j - i_{\sigma 2} + 1, \dots, j - i_{\sigma c(\sigma)} + 1$. (וכז הוא יספר התאמות של כל σ עבור כל ההיסטים השונים).

אלגוריתם 6 מספר-התאמות-2(T, P)

1. צור מערך I של מצביעים לרשימות הקשורות Σ .

2. עבור i מ-1 עד m בצע:

(א) הוסף את האינדקס i לרשימה הקשורת ב- $I[P[i]]$.

3. צור מערך M בגודל 1 $n - m + 1$

4. עבור j מ-1 עד n בצע:

(א) עבור i ברשימה הקשורת $I[T[j]]$

$M[j - i + 1] + 1 \rightarrow M[j - i + 1]$.i.

5. החזר את M

אם מט' המופעים של כל אחת בתבניות חסום במט' כלשהו c אז זמן הריצה הכלול למניות מופעים של כל האותיות יהיה בסה"כ nc . במקרה שלא נתון שום חסום $c = O(m)$ וסה"כ ניתן לחסום את זמן הריצה של האלגוריתם ב- $O(nm)$. קיבלונו אלגוריתמים שרצים בזמןים ($O(|\Sigma|nlogm)$ (הנאיבי) וכן $O(nm)$). נרצה לשלב אותם לאלגוריתם מהיר בהרבה – נחלק את האותיות בא"ב לשני סוגים לפי מט' המופעים של כל אחת בתבנית. לשם כך נעזר בערך סך c אותו נקבע בהמשך.

הגדרה: אותיות שמופיעות יותר מ-7 פעמים בתבנית יקרוו כבדות. אותיות שמופיעות פחות מ-7 פעמים בתבנית יקרוו קלות.

למה: מט' האותיות הכבדות בתבנית הוא לכל היוטר $\frac{m}{c}$.

כעת, נספר תחילה את התאמות שנוצרות בין המופעים של האותיות הכבדות, בעזרת האלגוריתם הראשון. סה"כ $O(|\Sigma|heavy n logm) = O(\frac{m}{c} n logm) = O(\frac{nm}{c} logm)$. אוח"כ נספר את התאמות שנוצרות בין מופעים של אותיות קלות, בעזרת האלגוריתם השני: סה"כ $O(nc)$.

אלגוריתם 7 מופר-ההאמות- i : (T, P) 1. צור מערך חדש M בגודל Σ 2. צור מערך C בגודל Σ 3. צור מערך I של מצבים לרישום מקשורות בגודל Σ 4. עבור i מ-1 עד m בצע:

(א) $C[P[i]] + 1 \rightarrow C[P[i]]$

(ב) הוסף את האינדקס i לרשימה המקושרת ב- $P[i]$ 5. לכל $\Sigma \in \sigma$ כך ש $C[\sigma] > c$:(א) ספור לכל היסט אפסרי של התבנית בטקסט את מספר ההתאמות של התו σ ע"י FFT של $T_\sigma \cdot P_\sigma^R$ ולוסף את ההתאמות למערך M 6. עבור j מ-1 עד n בצע:

(א) אם $C[T[j]] \leq c$

.ב. עבור i ברשימה המקושרת $:I[T[j]]$

$M[j-i+1] + 1 \rightarrow M[j-i+1]$

א.

7. החזר את M

זמן ריצת האלגוריתם יהיה $O(\frac{nm}{c} \log m + nc)$. נרצה לבחור ערך c שימזיר את הביטוי $\frac{m \log m}{c} + c$, כלומר $O(n \times \max\{\frac{m \log m}{c}, c\})$. נשים לב כי כאשר c גדול מואוד הגורם הדומיני הוא הימני, וכי אשר c קטן הדומיני הוא השמאלי. אנו מעוניינים בזמן אסימפטוטי - בעת שני הביטויים זהים:

$$\frac{m \log m}{c} = c \implies m \log m = c^2 \implies c = \sqrt{m \log m}$$

ומכאן, הזמן ריצת האלגוריתם הינו $O(n \sqrt{m \log m})$.

9 הרצתה 9: אלגוריתמים רנדומיים

9.1 מבוא והגדלה

הגדרה: יהיו אלגוריתם רנדומי A . אלגוריתם רנדומי הוא אלגוריתם שימושה בבחירה אקראית r כקלט (פרט לקלט והפלט הרגילים). יש לכך פירושים שונים - בקורס אלגוריתמים 1: אנחנו מניחים כי כל מספר ב- \mathbb{Z} הוא מספר אקראי שמתפלג באופן אחיד מטווח המספרים השלמים $[0, n]$ או $[1, n]$ או $[0, n]$ באשר n הוא גודל הקלט של האלגוריתם A .

דוגמה: אם הקלט שלנו הינו גרע $G = (V, E)$ אזי $|E| + |V| = n$ ונקבל שככל מספר ב- \mathbb{Z} הוא ממשי בטווח $[0, n]$ שמתפלג בו אחיד.

בעת שאנו מרכיבים אלגוריתמים דטרמיניסטיים, הנכונות והסבירויות ברורים ואפשרים להוכיחה. עם זאת, באלגוריתם רנדומי יכולים לקרות דברים שונים.

א. הריצה של אלגוריתם רנדומי A יכולה לדרוש זמן שונה. יתכן כי זמן הריצה יהיה תלוי בבחירה אקראית r .

ב. האלגוריתם A יכול להחזיר בסוף הריצה שלו תשובה שונה. כלומר, יהיו P_1, P_2 ריצות שונות של A . יתכן כי $P_1(A) \neq P_2(A)$.

משתנים מקריים:

א. זמן הריצה של אלגוריתם רנדומי יהיה משתנה מקרי שתלו依 ב- r .

ב. הנקנות של האלגוריתם (והצלהתו) היא גם כן משתנה מקרי שתלו依 ב α .

9.1.1 אלגוריתמי מונטה קרלו ואלגוריתמי לאס וגאס

הגדה: אלגוריתמי מונטה קרלו (*Monte – Carlo*) הם אלגוריתמים רנדומיים שזמן הריצה שלהם הוא דטרמיניסטי (ניתן לחסום אותו), אך הנקנות היא משתנה מקרי. לעומת זאת, האלגוריתם עלול לטעת ולא להצליח. נרצה לנתח את הסיכוי לטעות: שהיה כמה שיוטר קטן. (איך זה *Maybe correct*?).

$$\text{בהתנאי גודל קלט } n, \text{ נרצה כי הסיכוי לטעות} \geq \frac{1}{n^\alpha} \text{ עבור } 1 \leq \alpha \text{ קבוע.}$$

אם אכן הסיכוי לטעות} \geq \frac{1}{n^\alpha} \text{ אז אנחנו נאמר שהאלגוריתם צודק בסיכוי גובה} \leq 1 - \frac{1}{n^\alpha}.

הגדה: אלגוריתמי לאס וגאס (*Las – Vegas*) הם אלגוריתמים רנדומיים שזמן הריצה שלהם הוא משתנה מקרי, וננקנות היא דטרמיניסטי. לעומת זאת, האלגוריתם תמיד צודק אך זמן והירותו משתנה. באלגוריתמי לאס וגאס אנחנו נרצה לנתח את התוכנות החסתברויות של זמן הריצה. למשל: לחשב את תוחלת זמן הריצה, או חסם עליון לזמן הריצה בסיכוי גובה.

הגדה: אלגוריתמי אטלנטיק סיטי הם אלגוריתמים שוגם זמן הריצה וגם הנקנות הינם משתנים מקרים. (לא נתעסק בהם בקורס).

9.2 וידוא כפל מטריצות

קלטי: 3 מטריצות בינאריות מגודל $n \times n$. נסמן A, B, C .

פלט: לבדוק האם $C = A \times B$ באשר הטענו במודולו 2. (לא ביניاري, אלא כפל מעל \mathbb{Z}_2).

הערה. כפל במודולו 2 הכוונה היא שכפל הוא כמו AND וחיבור הוא כמו XOR . כאמור: $0+0=0, 1+1=0, 1+0=1, 0+1=1, 0 \times 1=0, 0 \times 0=0, 1 \times 0=0, 1 \times 1=1$

אלגוריתם נאייבי: נכפיל את A ב- B ונבדוק האם אכן הפלט הינו C . ראיינו כבר שההכפלה תעללה $O(n^\omega)$. וזה - לא ממש טוב לנו.

9.2.1 נסיוון ראשוני

נראה אלגוריתם מונטה קרלו שזמן הריצה שלו הינו $O(n^2)$ שטועה בסיכוי $\geq \frac{1}{2}$.

להלן האלגוריתם:

VERIFY-BINARY-MM-BASIC(A, B, C)

- 1 Pick a random vector $\vec{V} = (v_1, v_2, \dots, v_n)$ of bits
- 2 $\vec{V}_B \leftarrow B \cdot \vec{V}$
- 3 $\vec{V}_{AB} \leftarrow A \cdot \vec{V}_B$
- 4 if $\vec{V}_{AB} = C \cdot \vec{V}$
- 5 return "true"
- 6 else return "false"

האלגוריתם די ברור. הוא מבצע את המכפלה באמצעות וקטור ערכים אקראיים המורכב מביטים, בשני שלבים, ולאחר מכן בודק האם המכפלה הזאת שකולה למכפלה של המטריצה C בוקטור. אם כן: מחזיראמת, אחרת מחזיר שקר. נשים לב כי אם $C = AB$ אז בהכרח לכל וקטור \vec{v} יתקיים $C \times \vec{v} = AB \times \vec{v}$

נשים לב כי אם $C = AB$ האלגוריתם יהיה צודק תמיד, אך $C \neq AB$ אבל בחרה של וקטור לא טוב יוביל את האלגוריתם לטעות. מצב זה נקרא *false positive* (אמרתי נכון, בזמן שלא נכון). אם קיבלו לא מהאלגוריתם, בהכרח $C \neq AB$.

סיבוכיות זמן היריצה: נראה כי חישוב כל אחד מהשלבים 2 ו-3 עליה באופן נאיבי $O(n^2)$, כל אחד מהשלבים פולט וקטור בגודל n . שלב 4 מבצע שוב כפל שעולה $O(n^2)$. לבסוף הבדיקה האם V_{AB} שווה למינימום $O(n)$ שכך מעבר על n ערכי וקטור. סה"כ סיבוכיות האלגוריתם $O(3n^2 + n) = O(n^2)$

ניתוח הסיכוי לטעות:

נדיר $D = C - AB$. נניח $C \neq AB$. אז המטריצה $D \neq 0$. נסמנה $\{d_{ij}\}$ המשמעות הדבר: קיימים $d_{ij} = 1$ (פחות אחד שכזה).

הגדרה: נאמר כי וקטור \vec{v} הוא רע אם מתקיים $D \times \vec{v} = (C - AB) \times \vec{v} = 0$ (וקטור שכזה יגרום לאלגוריתם שלנו לטעות). אם \vec{v} אינו רע, אז נאמר כי \vec{v} הוא טוב.(Claim: אם $D \times \vec{v} \neq 0$).

למה 1: מספר הוקטורים הטובים הוא לפחות כמספר הוקטורים הרעים. (תחת הנחה ש $0 \neq D$)
מסקנה: אם בחרנו וקטור באקראי, והלמה אכן נכונה (מיד נוכח), אז הסיכוי לבחור וקטור טוב הוא לפחות $\frac{1}{2}$.

הוכחה: נתאר פונקציה חד-ערךית שmaps וקטוריים רעים לוקטוריים טובים. מכאן שבהכרח $|good-vectors| \leq |bad-vectors|$ לפי בדיחה, ואז סימנו את ההוכחה.
 יהי \vec{v} וקטור רע.(Claim: $\sum_{k=1}^n d_{\ell k} \times v_k = 0$).
 נזכיר כי קיימים $d_{ij} = 1$, כיון שהנחנו $D \neq 0$ (אם אכן אין סיכוי לטעות).
 נגדיר וקטור

$$w_j = \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

באשר הגדרת הוקטור הינה 0 פרט למיקום j (או $d_{ij} = 1$ בו יהיה 1).
 נזכיר. צאנו מנקודת הנחה ש- \vec{v} הינו רע (כלומר, הוא עבר עליון). נראה כי הוקטור $\vec{v}' = \vec{v} + \vec{v}_j$ הוא וקטור טוב.
 (Claim: $D \times \vec{v}' \neq 0$).

$$\vec{v}' = \begin{pmatrix} v'_1 \\ v'_2 \\ \vdots \\ v'_j \\ \vdots \\ v'_n \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_j \\ \vdots \\ v_n \end{pmatrix} + \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_j \\ \vdots \\ w_n \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_j + 1 \\ \vdots \\ v_n \end{pmatrix}$$

נסמן
נסתכל על $(D \times \vec{v})_i$. קיבל לפי הגדרה כי -

$$(D \times \vec{v}')_i = \sum_{k=1}^n d_{ik} v'_k = \sum_{k=1}^n d_{ik} (v_k + w_k) = \sum_{k=1}^n d_{ik} v_k + \sum_{k=1}^n d_{ik} w_k$$

נשים לב כי $\sum_{k=1}^n d_{ik}v_k = 0$ שווה לאפס, וטענו לגבי ℓ
 בפרט $i = \ell$ יתקיים.
 (Clomer):

$$(D \times \vec{v}')_i = \sum_{k=1}^n d_{ik}w_k = d_{ij}w_j + 0 + 0 + \dots + 0 = 1 \times 1 = 1$$

באשר כל שאר הערכים הינם אפס פרט ל-1, $d_{ij} = w_j$, וכן 1. סה"כ קיבלנו כי 1
 אבל $D \times \vec{v}' \neq \vec{v}$.
 סה"כ קיבלנו פונקציה:

$$f(\vec{v}) = \vec{v} + \vec{w}_j$$

נוכיח כי f חד חד ערכית.

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ \ddots \\ y_{n-1} \\ y_n \end{pmatrix}, X = \begin{pmatrix} x_1 \\ x_2 \\ \ddots \\ x_{n-1} \\ x_n \end{pmatrix}$$

נניח בשיילה כי f אינה חד חד ערכית. כלומר, קיימים

באשר $X \neq Y$. עבורם $f(X) = f(Y)$
 נסמן, $w_j = f(Y) = Y + w_j$, $X' = f(X) = X + w_j$, $y'_\ell = f(Y) = Y + w_\ell$. נשים לב כי הוספה או האזהה של w_j בקטור בסך הכל משנה את הביטוי h_j בקטור. מכאן שאם $X' = Y'$ אז לא ניתן כי $X = Y$. בסתיו. (בזה"כ מוסיפים בשתי הצדדים אותו דבר, וכך בהכרח נקבל $Y = X$, בסתיו). מסקנה: f חד חד ערכית.

9.2.2 נסיוון שני - אמפליפיקציה (Amplification)

הרעין הוא: נרץ את האלגוריתם הבסיסי k פעמיים. נתבונן באלגוריתם הבא:

VERIFY-BINARY-MM(A, B, C)

- 1 $k \leftarrow \alpha \log n$
- 2 **for** $i = 1$ to k
- 3 **if** VERIFY-BINARY-MM-BASIC(A, B, C) = "false"
- 4 **return** "false"
- 5 **return** "true"

אם באחת האיטרציות יוחזר לנו false מהאלגוריתם הבסיסי, אז האלגוריתם יחזיר שקר.(Clomer):
 בשביל שהאלגוריתם ייטה בכל האלגוריתם, ויזיר אמת למרות שלא מותקים $C = AB$ אז צריך
 לבצע k פעמיים וקטור רע.
נדיר את המאורע: = בכל איטרציה אנחנו בוחרים וקטור רע. נשים לב כי הסיכוי לטיעות בכל
 איטרציה תלוי תלוי באחרות. לכן נגידיר B כמאורע של הסיכוי לטיעות באיטרציה אחת.

$$Pr[A] = (Pr[B])^k \leq (\frac{1}{2})^k = \frac{1}{2^k}$$

נרצה כי זמן הריצה יהיה $\frac{1}{n^\alpha}$ עבור $2 \geq \frac{1}{2^k}$ כלומר $n^\alpha = 2^k$. זה יקרה באשר $\alpha \cdot k = log(n^\alpha) = \alpha log(n)$ ומכאן $\alpha log(n)$ נקבל כי ואכן אם נגדיר את k להיות $log(n)$

$$Pr[A] \leq \frac{1}{2^k} = \frac{1}{2^{\alpha log(n)}} = \frac{1}{n^\alpha}$$

סיבוכיות זמן הריצה: נראה כי הוללה מתבצעת k פעמים, בכל איטרציה קוראים לאלגוריתם הבסיסי שעליותו $O(n^2)$ ונקבל $O(\alpha n^2 log n) = O(n^2 log n)$.

9.3 מיון מהיר - Quick Sort: מבוא

בעיית המיון:

קלט: מערך $A = [a_1, \dots, a_n]$ של מספרים ממשיים (שונים - לא מהותי, מוחoti מתמטית מבחינה)
החוcharות).

פלט: מיון של A בסדר עולה.

מיון מהיר:

בחירה של איבר אקראי a_r , ויצירת *partition* סביב a_r . כל מה שמייננו יהיה גדול ממנו וכל מה שמושמלו יהיה קטן ממנו. ואת הצדימני והצד השמאלי, פוטרים איך לא: ברקורסיה. לאחר הרקורסיה קיבל את L ממוין, a_r בניהם ואת R ממוין ושה"כ נשרר את שני המרכיבים ונקבל את A ממוין.

באלגוריתם הקלاسي r נבחר באופן אקראי אחד בטוחה האינדקסים השלימים $[n, 1]$. חשוב שנשים לב - תמיד האלגוריתם מצליח למיין. מה שלא תמיד קורה: הוא זמן הריצה משתנה. ומכאן - מדובר באלגוריתם לאס וגאס.

נסמן ב- $T(n)$ את זמן הריצה על קלט בגודל n . במקרה הגרוע: $T(n) = 1$ או $n = r = 1$ ואז אנחנו צריכים לבצע מיון על קבוצה כלשהי בגודל $n - 1$ (בודאות L או G היא קבוצה ריקה). בשני המקרים הללו קיבל כי $T(n) = T(n - 1) + O(n) = O(n^2)$ כאשר הקלט קטן באחד, וכן נדרש עוד $O(n)$ לבייחו *partition* (האיחוד לבסוף). באופן כללי -

$$T(n) = T(|L|) + T(|G|) + O(n)$$

ומכאן כי במקרה הגרוע עלות האלגוריתם יכולה להגיע ל- $O(n^2)$.
מקרה טוב עבורנו - a_r הינו החציו של A . במקרה זה נקבל:

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + O(n) = 2T\left(\frac{n}{2}\right) + O(n) \in O(n log n)$$

כלומר: אם החלוקה תהיה תמיד בדיק בחצי, אז עלות זמן הריצה של האלגוריתם תהיה $O(n log n)$.

נשים לב כי - לא צריך חלוקה עד כדי כך טובה. גם אם החלוקת היא כזו שגודל כל צד הוא לפחות רבע מה, זמן הריצה יצא עדיין $O(n\log n)$. ויתר מה - אם כל צד הוא לפחות $\frac{n}{a}$ עבור a קבוע כלשהו, אז עדיין זמן הריצה יהיה $O(n\log n)$. הבדיקה זו, תוביל אותנו לגרסה מעט שונה של *Quick Sort*.

9.4 מילון מהיר פרנוואידי

האלגוריתם מבצע שינוי מואוד פשוט באלגוריתם המקורי: לאחר שמבצעים חלוקה, אם $|L| < \frac{n}{4}$ או $|G| < \frac{n}{4}$ אז האלגוריתם מבטל את בחירת *Pivot* הקיימת ומוחפש *Pivot* חדש. (באותה האופן).

זמן ריצה: אנחנו נרצה לנתח את תוחלת זמן הריצה, שכן זמן הריצה אינו קבוע. נרצה להציג שכל חשוב $T(n)$ קודם לכך נבעו מאינטואיציה, ואינם היו מדויקים. נרצה לחשב כתת תוחלת זמן הריצה שהיא הגורם המכريع באלגוריתמי לאס וגאס.
נזכיר בכלל התוחלת $E[X + Y] = E[X] + E[Y]$.

$$E[T(n)] = E[T(|L|) + T(|G|) + T(pivot)] = E[T(|L|)] + E[T(|G|)] + E[T(pivot)]$$

באשר $T(pivot)$ זה הזמן שלוקח למצוא *pivot* טוב. נשים לב כי $n - |G| - |L| = n - 1$. מכאן:

$$E[T(n)] = E[T(n - 1 - |G|) + E[T(|G|)] + E[T(pivot)]$$

נתמקד כתת ב $E[T(pivot)]$. נשים לב כי $T(pivot)$ הינו מס' הפעמים שמחפשים *pivot* כפול $O(n)$, שכן בכל חיפוש *Pivot* סורקים את המערך פעם אחת (בשביל למצוא מי מעליו ומהתחתי ולגלו).
את גDAL הקבוצות). נזכיר כי $E[aX] = aE[X]$ ומכאן שננסמן T : מס' הפעמים שמחפשים נקבע:

$$E[T(n)] = E[T(n - 1 - |G|)] + E[T(|G|)] + O(n) \times E[T]$$

בחירת *Pivot* היא ניסוי שמבצעים שוב ושוב, עד שמצליכים למצוא *Pivot* טוב. מספר הניסיונות עד ההצלחה מתפלג גאומטרית. וכך אם ההצלחה בכל ניסוי הינה p , אזי תוחלת מס' הניסיונות עד ההצלחה הינה $\frac{1}{p}$. נחשב כתת p . על מנת ש*Pivot* יהיה רע, הוא צריך להיות או ברבע האיברים הכי קטנים כי $\frac{n}{4} < |L|$. או ברבע האיברים הכי גדולים, וכי $\frac{n}{4} < |G|$. כלומר: בשביל ש*pivot* יהיה טוב הוא צריך להיות בטוח $[1, \frac{3n}{4}]$. נשים לב כי אלו בדיקות מהאפשרויות *pivot*.
מכאן המסקנה כי

$$p = Pr[Good - Pivot] = \frac{1}{2}$$

מכאן, $E[T] = \frac{1}{p} = \frac{1}{2}$.

$$E[T(n)] = E[T(n - 1 - |G|)] + E[T(|G|)] + 2 \times O(n)$$

נגדיר פונקציה $T'(k) = E[T(k)]$ ונסמן $x = |G|$. נקבל

$$E[T(n)] = T'(n-1-x) + T'(x) + 2 \times O(n)$$

כיוון שאנו יודעים כי $E[T(n)] = T'(n) = O(n \log n)$ נוכל לראות כי $\frac{n}{4} \leq x \leq \frac{3n}{4}$ $\iff -\frac{n}{4} \geq -x \geq -\frac{3n}{4} \iff \frac{n}{4} \leq x \leq \frac{3n}{4}$ ונקבל כי $n-1-\frac{n}{4} \geq n-1-x \geq n-1-\frac{3n}{4} \iff n-1-x \geq \frac{n}{4}-1 \geq n-1-\frac{3n}{4}-1$

$$E[T(n)] \leq T\left(\frac{3n}{4}-1\right) + T\left(\frac{3n}{4}\right) + 2O(n) \leq 2T\left(\frac{3n}{4}\right) + O(n)$$

והביטוי מימין חסום ב($O(n \log n)$ עם עץ רקורסיבי או סתם אינדוקציה שנחושך כעת. מש"ל.

9.5 תוחלת זמן הריצה של מילון מהיר "קלסטי"

בהתיחס לאלגוריתם הכללי של מילון מהיר, נרצה לחסום את תוחלת מס' השוואות. מתי האלגוריתם מבצע השוואות? האם האלגוריתם משווה בין כל זוג איברים? האלגוריתם לא משווה בין כל זוג איברים. נסהה להבין למה: האלגוריתם מקבל כקלט את המערך A ובוחר r . נניח כי בצד L ישנו a_i ובצד R ישנו a_j . איזו נשים לב: a_i ו- a_j ישו זה מול זה בעולם. אמורנו כי $A = (a_1, \dots, a_n)$. נסמן את הפלט y_1, \dots, y_n . (בחכרה מותקיים $y_1 < y_2 < \dots < y_n$). מתי האלגוריתם משווה בין y_i ל- y_j ? זה קורה מתי ש- y_i או y_j נבחרו כpivot ועדי לעצמו רגע, עברו כל $j \leq i \leq k$ אף אחד מהאיברים y_k לא נבחר להיות pivot. (אם נבחר - הם לא ישו). כל עוד לא נבחר y_k שכזה להיות pivot האיברים y_i ו- y_j יהיו יחד בקריה הרקורסיבית. אם בפעם הראשונה שהאלגוריתם בוחר איברים y_j, y_{j+1}, \dots, y_i , הוא בחרה של y_i או y_j איזי באותה בחרה האלגוריתם משווה בין y_i ל- y_j לבין y_j . נגידר משתנה מקרי ונסמן מאורע A : האלגוריתם משווה בין y_i ל- y_j .

$$X_{ij} = \begin{cases} 1 & A : \text{exist} \\ 0 & \text{o.w.} \end{cases}$$

$$E[T(n)] = (*) E\left[\sum_{i=1}^{n-1} \sum_{j=i+1}^n X_{ij}\right] = \sum_{i=1}^{n-1} \sum_{j=i+1}^n E[X_{ij}]$$

(*)-נשים לב כי זהה בבדיקה מס' השוואות.

$$E[X_{ij}] = 1 \times Pr[x_{ij} = 1] + 0 \times Pr[x_{ij} = 0] = Pr[x_{ij} = 1]$$

כפי שאמרנו, ההסתברות ש- $Pr[x_{ij} = 1]$ היא ההסתברות שתהיה השוואה בין y_i ל- y_j . אמורנו קודם שם ישו רק אם כל האיברים בינהם לא נבחרו להיות Pivot ואחד מהם נבחר. בטוחה ישנו $j-i+1$ איברים. מתוכם, שני איברים אס נבחרים ראשונים (y_i, y_j) יגררו ש- $X_{ij} = 1$ הבחירה כאן אקראית - יוניפורמת. הסיכוי ש- x_i או x_j יבחרו הראשונים בטוחה הינו $\frac{2}{j-i+1}$. מכאן $E[X_{ij}] = Pr[x_{ij} = 1] = \frac{2}{j-i+1}$

$$E[T(n)] = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{2}{j-i+1} = 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{j-i+1} =$$

$$2 \sum_{i=1}^{n-1} \left(\frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n-i+1} \right) \leq 2 \sum_{i=1}^{n-1} \sum_{\ell=1}^{n-j+1} \frac{1}{\ell} = 2 \sum_{i=1}^{n-1} \ln(n-i+1)$$

$$\leq 2 \sum_{i=1}^{n-1} \log(n) = O(n \log n)$$

כנדרש.

מסקנה: מיוון מהיר מבוצע בתוחלת $O(n \log n)$ השוואות.

9.6 מיוון דלי (*Bucket Sort*)

האלגוריתם הינו דטרמיניסטי. אם כן, נניח שהקלט מיוצר בצורה אקראיית. נפתר את בעיית המיוון כאשר כל מספר בקלט נבחר בהתפלגות אחידה בקטע $[0, 1]$. הרעיון יהיה שהאלגוריתם ייוון מובס על השוואות, וכך בתוכחת נציג לRDD($O(n \log n)$). עם זאת במקורה הכי גרווע נגע לסייעיות כו גם. האלגוריתם יעבד כך: נkeh את תוחת המספרים $[0, 1]$ וначלך אותו לא דליים. ככלمر $[0, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n}, 1]$ יש איבר נתחילה "לאורך" את האיברים אחד אחד. לכל דלי יכנסו מס' של איברים. בתוחלת: בכל דלי, יש איבר אחד. שכן יש n מספרים והסיכוי של כל איבר להכנס לתא הוא $\frac{1}{n}$ ליפול בתא מסוים. מכאן שתוחלת מס' האיברים בDALI תהיה $= 1 = \frac{1}{n} \times n$. אם תאורטית - בכל דלי נפל איבר אחד, אז קיבלו מיוון של האיברים שכן הדליים בסדר עולה. אם לא, ויתכן מצב תאורטי שבו נפלו הרבה בתא אחד, אז אנחנו בבעיה. פטודו לאלגוריתם יראה כך -

- **מיוון דלי** $((A = (a_0, \dots, a_n))$
- נכנסי כל איבר לדלי המתאים (האיבר נבחר מראש אקראיית!)
- נמיין כל דלי באמצעות מיוון הכנסה
- נשרשר את הדליים לפי סדרם

זמן הריצה:

נסמן $[i, \frac{i}{n}, \frac{i-1}{n}, \dots, \frac{1}{n}, 0]$ הדלי ה i . נסמן ב- n_i את מס' האיברים ב- B_i . נראה כי n_i הוא משתנה מקרי. כמו כן $n = \sum_i n_i$. נראה כי

$$\forall_{1 \leq i \leq n} : E[n_i] = \frac{1}{n} \times n = 1$$

להכנסי כל איבר לדלי המתאים עולה $O(n)$ זמן. שרשור הדליים גם כן עולה $O(n)$ זמן. נראה כי שלב השני, מיוון כל דלי באמצעות מיוון הכנסה, עולה לכל n_i $O(n_i^2)$ כי משתמשים במיוון הכנסה. מכאן, זמן הריצה הינו $O(n) + \sum_{i=1}^n n_i^2$ נסתכל על תוחלת זמן הריצה.

$$E[n + \sum_{i=1}^n n_i^2] = E[n] + \sum_{i=1}^n E[n_i^2] = n + \sum_{i=1}^n E[n_i]^2$$

נתקד בבאקט B_i וב n_i . נשים לב שהאלגוריתם לוקח n איברים וכל אחד מהם מצליח להכנס אל הבאקט B_i בסיכוי $\frac{1}{n}$. n סופר את מספר ההצלחות. לעומת: יש לנו ניסוי ש חוזר n פעמים עם סיכוי הצלחה $p = \frac{1}{n}$. ספירת מס' ההצלחות היא התפלגות בינומית ($n_i \sim Bin(n, \frac{1}{n})$). מכאן ש

$$E[n_i] = \frac{1}{n} \times n = 1$$

$$Var[n_i] = np(1-p) = 1 - \frac{1}{n}$$

זכור כי $Var[n_i] = E[n_i^2] - E[n_i]^2$ ומכאן נוכל לקבל כי

$$E[n_i^2] = Var[n_i] + E[n_i]^2 = 1 - \frac{1}{n} + 1^2 = 2 - \frac{1}{n}$$

סה"כ נחזור מעלה ונקבל

$$E[n + \sum_{i=1}^n n_i^2] = n + \sum_{i=1}^n E[n_i^2] = n + n(2 - \frac{1}{n}) = n + 2n - 1 = 3n - 1 = O(n)$$

כלומר, תוחלת זמן הריצה הינה $O(n)$.

נותרת השאלה: למה בחרנו במינון הכנסה? במערכות מניפולציות מתמטיות קל לטפל ב $\frac{2}{n}$ הרצה נוספת $M(n)$, כמו כן: בחרנו תאים מאד קטנים ואנחנו מוצפים שלא יהיה שם הרובה ערכיים. מינון הכנסה הוא המינון הכי טוב עבור קלטים מאד קטנים - הקבועים די קטנים. לעומת זאת, האלגוריתמים של $O(nlogn)$ מחזיקים קבועים גדולים ונוהים יעלים רק עבור n גדול.

10 הרצאה 10: שבירת סימטריה

10.1 מבוא והגדלת הבעיה

נניח כי יש לנו שני אנשים: אליס וbob. שניהם נמצאים בשיחות זום, ואצל שניהם המצלמות כבויות. כל אחד מהם רוצה לומר משהו אל הצד השני. למשל: אליס רוצה לומר לבוב שקוראים לה אליס, ובדומהו bob רוצה להגיד לאليس שלו שהוא bob. יותר מהה: יתכן שלשניים קוראים לאليس (לא בהכרח באותו שם שונה). אם שניהם ידברו בו זמנית, הם יעלו על הקול אחד של השני ולא יצלחו לשמעו את הקול. המטרה היא להציג שחק אחד "משדרת" קול.

נשים לב כי כל אלגוריתם דטרמיניסטי לא יכול לפתור את הבעיה. מדוע? מהו אלגוריתם דטרמיניסטי? אלגוריתם דטרמיניסטי הוא קוד כתוב שידוע מראש. וכך גם bob ואليس ישתמשו בקוד שנמצא במחשב שלהם בו זמנית, והוא יגיד להם לעשות אותו הדבר בדיק. מכאן שאנחנו חייבים להשתמש ברנדומות.

בעזרת מטבע רנדומי אצל כל אחד מה משתתפים ניתן להצליח. נניח כי נגידר את הטלת 1 להיות שהמשתתף מדבר 0 ושהמשתתף שותק. נראה כי במקרה של אליס ובוב מס' האפשרויות להטלת המטבע הינו:

00, 11, 01, 10

מצב טוב עבורנו הוא 10, 01. ומכאן מה הסיכוי להצלחה ושירותה משותף אחד בדיק שמדובר?

מבחן אפשר לקבל אלגוריתם: כל עוד אין הצלחה - הטל מטבע, אם יוצא אחד אז תשדר. נראה כי מס' הניסיונות עד להצלחה הראשונה הוא מסתנה מקרי שמתפלג גאומטרית, ולכן מס' הניסיונות תהייה $2 = \frac{1}{p} = \frac{1}{\frac{1}{2}} = 2^k$. ניתן גם לבדוק אחרי כמה סיבובים תהיה הצלחה בסיכוי גבוה. נראה כי כדי שלא תהיה הצלחה בא סיבובים ההסתברות הינה:

$$\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdots \cdot \frac{1}{2} = \frac{1}{2^k}$$

עבור $1 < c$ קבוע, נקבל כי הסיכוי לא להצלחה בא ניסיונות הינו:

$$\frac{1}{2^k} = \frac{1}{2^{c \log n}} = \frac{1}{n^c}$$

כלומר, הסיכוי לא להצלחה קטן פולינומית. ומכאן שיש הצלחה בסיכוי די גבוה.

אפשר להרחיב את הבעיה, מה אם יש שלושה אנשים וANO רוצחים לשבור סימטריה? נרצה שכל אחד ינסה לשדר בסיכוי שלישי. בשביל שתהיה הצלחה בניסוי אחד: צריך שאחד ישדר, ושניים אחרים ישתקו. הסיכוי לכך (מתפלגBINOMIOT) הינו:

$$\binom{3}{1} \times p \times (1-p)^2 = 3 \times \frac{1}{3} \times \left(\frac{2}{3}\right)^2 = \frac{4}{9}$$

ומכאן שבתוחלת לאחר $\frac{1}{\frac{4}{9}} = \frac{9}{4} = 2.25$ סיבובים תהיה הצלחה.

וכמובן איך לא, מה קורה כאשר ישנים n שחקנים? נשים לב (גם כשייש 3 שחקנים) שהשחקנים זוקקים לדעת מהו n . נניח כי כל אחד מהם מנסה לשדר בסיכוי p . הסיכוי לשידור (בחירה מוגבלת) הינו:

$$Pr[Success] = \binom{n}{1} \times p \times (1-p)^{n-1}$$

נרצה למקסם את הסיכוי להצלחה, ככלומר p . מכאן שהסיכוי להצלחה המקסימלי יתקבל כאשר $p = \frac{1}{n}$ (גירה פשוטה מראה זאת) מכאן שהסיכוי להצלחה הינו:

$$Pr[Success] = \binom{n}{1} \times \frac{1}{n} \times \left(1 - \frac{1}{n}\right)^{n-1} = \left(1 - \frac{1}{n}\right)^{n-1} = \frac{\left(1 - \frac{1}{n}\right)^n}{1 - \frac{1}{n}} \leq \frac{e^{-1}}{1 - \frac{1}{n}} =$$

$$\implies \Pr[\text{Success}] \leq \frac{n}{e(n-1)} \leq \frac{1}{e}$$

ולכן תוחלת מס' הניסיונים עד להצלחה תהיה **בערך** $\frac{1}{\frac{1}{e}} = e$

אם נגידו **פורמלית** את הבעיה:

קלט: n שחוקנים.

פלט: מנהיג יחיד.

10.2 המודל המבוזר המקומי

נניח שיש n מוחשבים בראש מוחשבים, כל קודקוד יונגן מחשב וישנו קשיות בין מוחשבים (גרף). כל קשת היא חיבור רשות בין מוחשבים. לכל מוחשב יש כוח חישוב איסופי. לעומת זאת - אנחנו נניח שכל מוחשב באfonו מיקומי יכול להיזכר אלגוריתמים מואוד מוסככים שזעמו הרויה של הספmers מואוד גכו: בחוויה, כלוור, תאורתית, כל מחשב יוכל לפתח בעיות NP קשות בשבייה. בכל וזרת זעמו, ככל יונגן יכול לשולח הודעות גדולות כראינו לכל אחד משוכיו (זה יעלה סיבוג אחד של תקשורת). נראה כי בזעמו שקדקוד אחד שליח הודעה לשוכיו, גם שאור הקודקודים שלוחה הודעה לשוכיו. לעומת זאת, השולחות הודעות מותכעת במקביל. בפועל זהה, נძוז את הבעיות של האלגוריתמים שלו באמצעות מס' סכבי התקשרות. **למשל, בהינתן גраф כנייל:**

$$a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow \dots \rightarrow a_d$$

אנו נאלץ **למ' סכבי תקשורת**, בסכוב הראשון ההודעה תצא מ- a_1 אל a_2 , בסכוב השני מ- a_2 אל a_3 וכו' הלאה. אנו לו ברורה אחרות כאו כיוון שאנו לו זרכי קוינור. חלק זה של הסпрос: נסמן כה את מס' הקודקודים, ווניה כי **כל מוחשב יודע את a .** לכל מוחשב אין **id** (יש **id** אם מושך שימוש ב-*random*). נפרט:

הגדרה: המודל המפואר המקומי הוא גראף לא מכוון ($G = (V, E)$) = כל קודקוד הוא מחשב שמרץ קלט מקומי. זמן מחולק ל一步步ים, בכל סכוב כל קודקוד יכול לשולח הודעה גדולה כרצונו לכל שכני. כל קודקוד מכיר את שכני. בעולם הדטרמיניסטי מינימום כדי לכל קודקוד ישנו **id**, עם זאת במודול המפואר לכל קודקוד אין **id**. סכוב הוא חישובי מקומי פולינומי, שיש בו שליחה וקבלת הודעות. זמן הריצה יהיה כמספר ה一步步ים. **כל קודקוד מרץ מראש את אותו האלגוריתם.**

במודול המבוזר המקומי, ישנו שתי בעיות של שבירת סימטריה שיויכלות לעניין אותו. **צביעה:** המטרה היא לצבעו את קודקודי הגראף (لتת מספרים מהם צבעים) כך שלכל קשת שני הקודקודים צבועים שונים. נשים לב שלא רנדומיות לא ניתן לפטור את הבעיה, שכן יתכן גראף סימטרי עם שני קודקודים a_2, a_1 . לכל אחד מהם יש שלושה שכנים נוספים. אז מבחינת כל קודקוד יש אותו, יש לו שלשה שכנים ויש קודקוד נוסף שיש קשת בינם גם לו יש שלושה קודקודים. אין שניי בינם ולכן הם יבצעו את אותה החלטה באלגוריתם דטרמיניסטי.

מציאת קבוצה בלתי תלוי מקסימלית: בהינתן גראף רצוה לבחור תת קבוצה של הקודקודים שהיא:
 1. **מקסימלית** (במובן הילוקאלי): לומר לא ניתן להווסף עוד קודקוד ולהשאר בקבוצה בת"ל
 2. אין זוג שכנים שנבחר.

במדעי המחשב לרוב מדברים על קבוצה בלתי תלויה מקסימומית (מקסימום), זו בעיה שנייתן לפתרו.

כאן נרצה למצוא את הקבוצה הבלתי תלויה בגודל הכיכי גדול. זו בעיה הרבה יותר קשה.

10.3 בחירת ID במודול המפוזר

כרגעון ראשוני, נניח כי כל קודקוד בוחר ID מהטוווח $[n]$ באקראי, מה הסיכוי שיש התנגשות? נקבע $[n] \in i$ ונגיד n קבוצת כל הקודקודים שבחרו את i .

$$Pr[n_i = 0] = \binom{n}{0} \times \left(1 - \frac{1}{n}\right)^n = \frac{1}{e}$$

נבחן כי הסיכוי ש- $n_i = 0$ גורר כי כל תא קיבל בדיקת כדור אחד, שכן נניח כי יש תא עם יותר משני כדורים איזי בהכרח יהיה תא עם אפס. מכאן שהסתוכי שאין התנגשות הוא $\frac{1}{e}$.

$$Pr[\exists_{i \in [n]} n_i = 0] = Pr[\bigcup_{i \in [n]} \{n_i = 0\}] \leq \sum_{i=1}^n Pr[n_i = 0] \approx \frac{n}{e}$$

ונבחן כי מדובר ביחסם די גרווע. ככלומר הסיכוי שיש התנגשות (תא אחד עם אפס) חסום לנו כאן ב- $\frac{n}{e}$, נרצה לשפר. נבחן כי הסיכוי שקיים תא עם אפס הוא:

$$Pr[] = \frac{n^n - n!}{n^n} = 1 - \frac{n!}{n^n}$$

כלומר, הסיכוי הזה די גדול שכן $n > n^n$ ולכן הסיכוי שיש id שנבחר פעמיים (התנגשות) יחסית גדול.

עתה, נבחר id באקראי מטווח טווח $[n^c]$. עבור $c > 0$ פרמטר. נבחן כי הסיכוי להתנגשות הינו:

$$Pr[Id(u) = Id(v)] = \frac{1}{n^c}$$

$$Pr[\exists u \neq v : Id(u) = Id(v)] = Pr[\bigcup_{u \neq v} Id(u) = Id(v)] \leq \sum_{u \neq v} Pr(Id(u) = Id(v)) = \binom{n}{2} \times \frac{1}{n^c} < \frac{n^2}{n^c} = \frac{1}{n^{c-2}}$$

ולכן, עבור $c \geq 3$ קיבל כי הסיכוי שלא תהיה התנגשות, גבוהה מאוד.

10.4 בעיית הצביעת

קלט: גרף $G = (V, E)$

פלט: פונקציית צביעת $C : V \rightarrow [1, \dots, c]$ כך שהצביעת חוקית (לכל $(u, v) \in E$ מתקיים $c(u) \neq c(v)$ וכן c מינימלי).

זכור כי בגרף דו צדדי ניתן תמוך אותו בשני צבעים, גרפ' תלת צדדי ניתן לצביעה בשלושה צבעים. גרפ' א-צדדי ניתן לצביעה בא-צבעים. (שכן די ברור הרעיון אין קודקודים בתוך כל צד אז אפשר לצבוע כל צד בא-צבעים שונים).

באופן כללי, הקושי הוא במצבה המספר הקטן ביותר של צבעים שביהם ניתן לצבוע את הגרף באופן חוקי. מדובר בעיה מאוד קשה. לא ניתן לפתור אותה באופן דטרמיניסטי בפחות מ($O(2^n)$). ואך - מאמינים כי לא ניתן להגעה לזמן פולינומי. אם כן, יש משפחות של גרפים שנitin לצבוע אותם באופן פולינומי. גרפ' דו צדדי למשל.

נסמן ב Δ את הדרגה הכי גדולה בגרף. כמובן, לכל קודקוד $V \in \mathcal{V}$ ישנה דרגה $\Delta \leq \deg(v)$.
הנחה: ניתן לצבוע את הגרף באופן חוקי + 1 צבעים. מדווק' נתחל בקודקוד מדרגה d כלשהו, בהכרח $\Delta \leq d$, גם אם $\Delta = d$. הוא משיק ל Δ קודקודים שככל אחד מהם תפס צבע אחר, במקרה הגרען ביותר שכן $d = \Delta$ עדין הוא יכול להשתמש בצבע האחרון. באופן כללי הוא יוכל להשתמש ב 1 – $d = \Delta + 1 - d \geq \Delta + 1 - \Delta = 1$ צבעים.

10.5 צביעה במודל המבוזר המקומי

נניח שיש לנו קודקוד v ויש לו שכנים v_1, v_2, \dots, v_k . הבעיה של הקודקוד v הוא שהוא לא יודע כיצד שכניו פועלים. למשל: אם קודקוד v היה יודע שכינוי אינם שכנים אחד של השני – איזי קודקוד v היה רוצה לומר להם: תצבעו כולכם באותנו הצבע, ואני אצבע את עצמי בצבע שונה משלכם. כרגע בסיסי מאד – אני יכול להחליט שזיהו את הקודקוד v של השודעה (v_k, v_1, v_2, \dots, v) לכל שכני. כמו כן: ככל ששלח הודעה לכל השכנים שלו עצמו. ואז – אני מקבל את ההודעות של כל שכני, ואוכל להסתכל האם באחת ההודעות אני מזוהה קודקוד שכבר יש לי. כמובן: האם השכנים של הם גם שכנים אחד של השני. נשים לב שקצת רימינו – שכן טענו כי לקודקודים אין id , אז איך נוכל לשולח הודעה שכזו? דבר על תהליך בחירת ID שקרה במקביל עבור כל הקודקודים.

בחירת ID :

1. כל קודקוד בוור id בין המספרים $[1, 2, \dots, n^{c+2}]$ עברו c קבוצה.
2. שלח ID לכל השכנים. (שכל אחד ידע את id' של השכנים שלו)

נרצה לבדוק מה הסיכוי שישנם שני קודקודים עם אותו id .

$$Pr[sameID] = 1 \times \frac{1}{n^{c+2}} = \frac{1}{n^{c+2}}$$

נסמן ב A_{uv} את המאורע שבו u ו- v בחרו את אותו Id . מכיוון $Pr[A_{uv}] = \frac{1}{n^{c+2}}$. נסתכל על המאורע הבא, שימושיתו שאף אחד לא בחר את אותו id .

$$Pr[\overline{\bigcup_{u,v \in V} A_{uv}}] = 1 - Pr[\bigcup_{u,v \in V} A_{uv}]$$

נראה כי

$$Pr[\bigcup_{u,v \in V} A_{uv}] \leq \sum_{u,v \in V} Pr[A_{uv}] \leq n^2 \times \frac{1}{n^{c+2}} = \frac{1}{n^c}$$

וקיבלנו כי

$$Pr[\overline{\bigcup_{u,v \in V} A_{uv}}] = 1 - Pr[\bigcup_{u,v \in V} A_{uv}] = 1 - \frac{1}{n^c}$$

ולכן הסיכוי לטעות קטן פולינומי, והסיכוי להצלחה גדול מאוד פולינומי.

סה"כ קיבלנו כי באמצעות טכניקה רנדומית פשוטה, יצרנו לכל קודקוד ID . מכאן: יש ממשמעות לשילוח הדעה לכל השכנים של השמות השכנים.
נראה כי ישנו קושי - בהחלט יכול להיות שאפילו שלשכנים של אין קשרות בינם, עדין לא ניתן לצבוע את כל השכנים באותו צבע. הרבה דוגמאות יכולות להיעיד על כך: הרעיון הזה פשוט מדי, וחשוב לוקליות מקומית אך הנגרף יתר גדור מזה. בחירה מקומית יכולה להשפיע על הנגרף כולו באופן שלא ציפינו. יתרה מזאת - כל מעגל אי זוגי דורש לפחות 3 צבעים. ושיטה זו בפועל תניב 2 צבעים. וכן: יש קושי להבין האם קודקוד נמצא במעגל אי זוגי. שכן תיכון כי גודל המעגל האי זוגי מאוד גדול ויקח הזמן זמן והודעות להבין שאנו נמצאים בכך.

אנו נרצה לצבוע את הנגרף באמצעות 2Δ צבעים. (ראינו כבר כי ניתן להשתמש ב- $1 + \Delta$ צבעים, אך המטרה באלגוריתם שנראה בהרצתה הוא לא משווה חדשי - אלא להבין את המודול המבואר המקומי)

10.6 אלגוריתם צביעת

כעת נתאר את האלגוריתם שיצבע את הנגרף באמצעות 2Δ צבעים.
 1. נבחר צבע באקראי מבין $[1, \dots, 2\Delta]$. נשים לב - ישנה כאן הנחה סמייה: כל קודקוד $V \in v$ מכיר את Δ .
 2. נשווה עם השכנים. ישנו שני מקרים -
 א. אף שכן לא בחר את הצבע שאנו בחרנו: במקרה זה, אנחנו הצלחנו. נחליט שזה הצבע שלנו.
 ב. אחרת, קיים לפחות אחד שבחר את הצבע שאנו בחרנו, במקרה זה אנחנו נשנה לבחור שוב את הצבע. (נזהר לו!).
 נראה את האלגוריתם עבור קודקוד יחיד $V \in u$ -

```
Color ( $\Delta$ ):
while(true):
    -pick random color from  $[1, \dots, 2\Delta] \rightarrow c$ 
    - send  $c$  to neighbors
    - recieve colors of neighbors
    -if there is no neighbor with color  $c$  so return and update  $C(u) = c$ .
```

נשים לב שבבדיקה אנחנו בודקים את כל השכנים של הקודקוד, ולא רק את אלו ש"אקטיביס".
 כרגע. קלומר - בודקים גם את אלו שסימנו לצבוע את הקודקוד שלהם.

10.6.1 נכונות האלגוריתם

מה הסיכוי שהבדיקה בשורה 5 תצליח? קלומר: אין שכן שם בחר את הצבע c .
 לכל קודקוד יש דרגה $d \geq \Delta$ ולכן לא משנה איזה צבעים השכנים בחרו, תמיד יש לפחות $d - 2\Delta$ צבעים פנויים. שורה 5 בהכרח מצליחה אם הצבע c שנבחר הוא אחד מהצבעים הפנויים. נסמן ב- x את מס' הצבעים הפנויים ברגע זה. בהכרח $x \geq 2\Delta - d$.

$$Pr[SuccessLine5] = \frac{x}{2\Delta} \geq \frac{2\Delta - d}{2\Delta} = 1 - \frac{d}{2\Delta} \geq_{\Delta \geq d} 1 - \frac{\Delta}{2\Delta} = \frac{1}{2}$$

כלומר, הסיכוי להצלחה בשורה 5 הוא גדול שווה ל- $\frac{1}{2}$.
icut, נרצה לחסום את מס' היסודות שהאלגוריתם עולה עד לצביעה חוקית של כל הגראַף.

נסמן ב- V_i את קבוצת הקודוקודים שעדיין לא נקבעו אחרי i איטרציות של האלגוריתם. בהכרח
לפי הגדרה $V = V_0$. נראה כי

$$\forall u \in V : Pr[u \in V_i] \leq \frac{1}{2^i}$$

כיוון שהסיכוי שקודוקוד יהיה בקבוצה, איי בכל האיטרציות הקודמת היה כשלון בשורה 5. שcn
אנו יודעים שכשלון בשורה 5 קטן שווה מסיכוי חצי.

$$E[|V_i|] = \sum_{u \in V} Pr[u \in V_i] \leq \frac{n}{2^i}$$

שcn $= |V| = n$. לכן, אחרי 1 איטרציות קיבל כי

$$E[|V_{logn+1}|] \leq \frac{n}{2^{logn+1}} = \frac{1}{2}$$

כלומר מס' הקודוקודים בתוחלת לאחר $logn + 1$ איטרציות הוא חצי. נראה כי ישנה טעות
נפוצה בשלב זה: להגיד מכאן ובע Ci מס' האיטרציות עד שכל הקודוקוד צבעים הוא לכל היותר
.”. נראה כי זה שהთוחלת היא לכל היותר חצי (לא יתכן הרי חצי איבר), לא אומר שלאחר
 $logn + 1$ איטרציות הקבוצה תהיה ריקה (אין כאן לינאריות למשה, זה שיש חצי איבר שנשאר זה לא
גורר שלאחר $1 + log(n)$ איטרציות נסימן. התוחלת למשה אכן אינה פונקציה הופכית ולכן הכוון
ההפוך לא נכון). בהתחשב בתובנה זו: כיצד ממשיכן? נראה כי תמיד יתקיים לפי האלגוריתם
וההסתברות שראינו קודם קודם כי -

$$E[|V_i|] \leq \frac{1}{2}|V_{i-1}|$$

נזכור באז שווין מרקוב. שאומר את הטענה הבאה: $Pr[X \geq t] \leq \frac{E[X]}{t}$. מכאן, נרצה לחשב מה
הסיכוי שגודלו של קבוצה יהיה גדול שווה ל- $\frac{3}{4}|V_{i-1}|$.

$$Pr[|V_i| \geq \frac{3}{4}|V_{i-1}|] \leq \frac{E[|V_i|]}{\frac{3}{4}|V_{i-1}|} \leq \frac{\frac{1}{2}|V_{i-1}|}{\frac{3}{4}|V_{i-1}|} = \frac{2}{3}$$

מכאן שסיכוי זה הוא לכל היותר $\frac{2}{3}$. ומכאן:

$$Pr[|V_i| < \frac{3}{4}|V_{i-1}|] = 1 - Pr[|V_i| \geq \frac{3}{4}|V_{i-1}|] \geq 1 - \frac{2}{3} = \frac{1}{3}$$

כלומר, הסיכוי שבאייטרציה ה- i הצלחנו לפחות לפחות רביע מתקודודים שלא היו צבעים קודם
באייטרציה ה- $i-1$ היא לפחות סיכוי של $\frac{1}{3}$. המספר שלישי הוא קבוע, וזה מה שהוא חשוב לנו. נסמן $p = \frac{1}{3}$.

נאמר שאיטרציה היא ”טובה“ אם היא הצלחה לפחות לפחות רביע מתקודודים שלא היו צבעים
בתחילת האיטרציה: ככלומר $|V_i| \leq \frac{3}{4}|V_{i-1}|$ (כל היותר נשארו $\frac{3}{4}$ מכמה שהוא פעם קודמת). ורעה=לא
טובה.

בහינתן יהיו k איטרציות טובות, נשארנו עם $n \geq (\frac{3}{4})^k$ מוקודדים. נרצה כי:

$$(\frac{3}{4})^k \times n < 1 \implies n < (\frac{4}{3})^k \implies \log_{\frac{4}{3}}(n) < k$$

כלומר אם $k = \log_{\frac{4}{3}}(n) + 1$ אי הצלחנו לצבע את כל הגרף. ככלומר אם נkeh כנ"ל כמס' האיטרציות הטובות אנחנו סימנו. נניח: החישוב באנו הינו דטרמיניסטי לחולוין, שכן אם היו k איטרציות טובות, נשארנו עם לכל היותר $n^{(\frac{3}{4})^k}$ קודוקודים, לכן קיבלנו חסם על k ובפרט את k .

קודם לכן ראיינו כי $\Pr[|V_i| < \frac{3}{4}|V_{i-1}|] \geq \frac{1}{3}$, ככלומר הסיכוי שההיא איטרציה טובה לפחות. לכן, בתחילת (התפלגות גאומטרית) עם $\frac{1}{3} \geq p$ מס' האיטרציות שיש בראץ עד שמקבלים איטרציה טובה ראשונה הוא $\frac{1}{p} \leq 3$. **ומכאן:** מס' האיטרציות הטובות הינו $\log_{\frac{4}{3}}(n) + 1$, כפול 3 מס' האיטרציות הקשורות עד שמקבלים את האיטרציה הטובה הבאה (הסתברות היא $\frac{1}{3}$ וזה משתנה גאומטרי). סה"כ נקבל כי

$$3\log_{\frac{4}{3}}(n) + 3$$

הוא תוחלת מס' האיטרציות עד שאין קודוקודים לא צבועים. ואכן, מס' האיטרציות שהאלגוריתם יעשה תהיה $O(\log n)$.

כעת, נרצה גם לבדוק את הנכונות במקרה הגרוע ביותר. ראיינו כי

$$E[|V_i|] \leq \frac{n}{2^i}$$

נרצה לבדוק מה הסיכוי שלא סימנו עבור $i = c \log n$ עבור $c > 1$ קבוע.

$$\Pr[|V_{c \log n}| < 1] = 1 - \Pr[|V_{c \log n}| \geq 1]$$

$$\Pr[|V_{c \log n}| \geq 1]_{markov} \leq \frac{E[|V_{c \log n}|]}{1} \leq \frac{n}{2^{c \log n}} = \frac{1}{n^{c-1}}$$

ונקבל כי

$$\Pr[|V_{c \log n}| < 1] = 1 - \Pr[|V_{c \log n}| \geq 1] \geq 1 - \frac{1}{n^{c-1}}$$

כלומר, הסיכוי שלא סימנו קטן פולינומי. וכך הסיכוי להצלחה יחסית טוב.

הערה. מדובר באלגוריתם לאס וגאס כי הוא תמיד צודק ומצילח. בתחילת, חישבנו את זמן הריצה שלו בתחילת.

10.7 בעיית אוסף הקופונים (איסוף מדבקות לאלבום - תרגול)

נתאר את הבעיה הבאה: ישנו n סוגים של קלפים - למשל קלפי אלבום סופרגול. מכל אחד מסווג הקלפים מיוצרים איסוף כרטיסים. מטרתנו היא למלא אלבום, אשר מכיל מקום אחד לכל אחד מסויי המדבקות. ישנה עירימה שnitן ללחשת מנתה כרטיס, וכל כרטיס הוא מאחד מהסוגים בהתקלויות אחת. כמה קלפים יש לשאזר מהעירימה עד שנחזיק בידינו לפחות עותק אחד מכל סוג?

10.7.1 תוחלת מספר הכרטיסים שיש להוציא עד הוצאה כל הסוגים

נסמן ב- X את מספר הקלפים שיש להוציא מהחביליה עד שנחזיק קלפים מכל הסוגים. לכל $1 \leq i \leq n$ נסמן ב- X_i את מס' הקלפים שונים מרהגע שיש בידינו $- i$ סוג קלפים, עד הרגע שיש בידינו i סוג קלפים. נבחן כי בהכרח $\mathbb{E}[X_1] = \frac{n}{n-i+1}$, כאשר יש בידינו $- i$ סוג קלפים, בכל משיכה הסיכוי שנקלב קלף מסווג חדש הוא $\frac{1}{n-i+1}$. (רוצחים מתוך n הקלפים שיצאו קלפים שאיןם ה- i הראשונים). מדובר במשתנה גאותרי, ולכן תוחלת מס' הקלפים שיש להוציא עד להוצאה קלף חדש היא $\mathbb{E}[X_i] = \frac{n}{n-i+1}$, אם כן נקבל:

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^n X_i\right] = \sum_{i=1}^n \mathbb{E}[X_i] = \sum_{i=1}^n \frac{n}{n-i+1} = n \sum_{i=1}^n \frac{1}{n-i+1}$$

אם כן, נסמן $j = n - i + 1$ ונקבל

$$= n \times \sum_{j=1}^n \frac{1}{j} = nH_n \approx nlnn$$

לכן, תוחלת מס' הקלפים שיש להוציא עד להוצאה לפחות קלף אחד מכל סוג הינה $O(nlnn)$.

10.7.2 חסם עליון על מס' הקלפים שנצטרכז להוציא בהסתברות גבוהה.

יהי X המשתנה המקרי המיצג את מספר הקלפים שנוציא עד שנראה קלף אחד לפחות מכל סוג. נסתככל על קלף i הסיכוי שנוציא את הקלף בכל אחד מחשבבים היה $\frac{1}{n}$. לכן הסיכוי שלא נוציא אותו היה $1 - \frac{1}{n}$. הסיכוי שבסמוך $knlnn$ סיבובים לא נוציא את הקלף הינו:

$$(1 - \frac{1}{n})^{knlnn} \leq (\frac{1}{e})^{knlnn} = \frac{1}{n^k}$$

נסמן ב- X_i את המאורע שאחריו $knlnn$ סיבובים טרם הוציאנו את קלף מס' i . המאורע שלא הוציאנו איישחו קלף כלשהו במהלך $knlnn$ סיבובים הוא היחיד של כל X_i עבור $n \leq i \leq 1$ ולכן לפי חסם האיחוד:

$$Pr[\bar{X}] \leq \sum_{i=1}^n Pr[X_i] = \frac{1}{n^k} \times n = \frac{1}{n^{k-1}}$$

לכן, נגדיר $c = k + 1$ ונקבל כי הסיכוי שהוציאנו את כל סוג הקלפים בתוך $(c+1)nlnn$ סיבובים הוא לפחות $\frac{1}{n^c} = 1$, סיכוי גבוהה מאוד. נבחן כי למעשה ביצענו הוכחת נכונות "זמן הריצה שגילינו". מדובר באלגוריתם לאס וגאס - תמיד צודק, אך זמן הריצה משתנה מקרי.

10.8 קבוצה פוגעת (*Hitting – Set*)

הגדירה. נתון עולם של U איברים. ומספר $n < R$. בנוסף נתנות $k < n^{c_1}$ (כאשר c_1 קבוע כלשהו) קבוצות:

$$S_1, \dots, S_k \subseteq U$$

כך שלכל $i \in [k]$ מתקיים $|S_i| \geq R$ קבוצה $A \subseteq U$ קבוצה פוגעת, אם לכל $i \in [k]$ מתקיים $\emptyset \neq A \cap S_i$. (כלומר, היא לא זרה לאף אחת מ- k הקבוצות).

ביעית מציאת קבוצה פוגעת בגודל מינימלי היא בעית NP קשה - אך לא סביר שנצלה לפטור אותה בזמן פוליאומי. במקומות זה, נראה כיצד למצוא קבוצה פוגעת בגודל קטן יחסית, בຄלות רבה ובמהירות. נוכחים עת שנייה למצוא בהסתברות גבוהה קבוצה פוגעת בגודל $O(\frac{n}{R} \log n)$.

משפט 2. בהינתן k קבוצות $S_1, \dots, S_k \subseteq U$, $|S_i| \geq R$, קיימים אלגוריתם המוצא קבוצה פוגעת בהסתברות טובה של $O(\frac{n}{R} \log n) - 1$ בגודל $\frac{1}{n^c}$.

נבחן, כי אם הקבוצות S_1, \dots, S_k זרות בגודל בדיק R המוחות חילוק של U , אז בהכרח $k = \frac{n}{R}$ כך שהגודל המבוקש מחייב קבוצה של $\frac{n}{R}$ איברים לקבוצה פוגעת, וזה פוגע בגודל הנדרש בפקטור של $O(\log n)$. כמו כן, גודל הקבוצה הפוגעת המובטח במשפט כולל איינו תלוי ב- k . כלומר הקבוצה שנבחר צפופה לפחות בפחות R ביחס לגבולה העליון.

נבחר את הקבוצה באופן דומה לבעית אוסף הסופרגול. נבעצ' $(n)^{c_2 \frac{n}{R} \ln(n)}$ בחירות אקראיות של איברים, כל בחירה מגילה אחד מ- n האיברים ב- U ללא שום תלות ביןיהם בבחירה (בחירה עם חירות), וכן ברור שגודל הקבוצה שתתקבל הינו $O(\frac{n}{R} \ln(n))$. נבחן כי יתכו נבחר כפולות של אותו איבר כך שגודל הקבוצה לא חייב בדיק $c_2 \frac{n}{R} \ln(n)$. תהי S_i קבוצה כלשהי, כיון שמתקיים $|U| = n \geq |S_i| \geq R$ בבחירה של איבר אקראי הסיכוי שפגע בקבוצה הינו לפחות $\frac{R}{n}$. לכן, בבחירה של איבר אחד הסיכוי שלא נפגע ב- S_i הוא לפחות $1 - \frac{R}{n}$ ולכן בבחירה של $c_2 \frac{n}{R} \ln(n)$ איברים הסיכוי שאיבר לא יפגע בקבוצה הינו לפחות $(1 - \frac{R}{n})^{c_2 \frac{n}{R} \ln(n)}$:

$$(1 - \frac{R}{n})^{c_2 \frac{n}{R} \ln(n)} \leq e^{-c_2 \ln(n)} = \frac{1}{n^{c_2}}$$

אם כן, כיון שם' הקבוצות k פוליאומי ב- n דהיינו $k \leq n^{c_1}$, נבחר $c_2 = c + c_1$ ונקבל לפי חסם האיחוד שהסיכוי שתהיה איזושהי קבוצה שלא פגעה בה הינו:

$$k \times \frac{1}{n^{c_2}} \leq \frac{1}{n^{c_2}} \times n^{c_1} = n^{c_1 - c_2} = n^{-c} = \frac{1}{n^c}$$

כלומר, בהסתברות די גבוהה $1 - \frac{1}{n^c}$ אנחנו נפגע בכל הקבוצות.

הערה. מדובר באלגוריתם מונטה קרלו: זמן הריצה תמיד אותו דבר, אך הנכונות היא משתנה מקרי.

10.9 ערבות אחיד (תרגום)

נזכר בבעית המזכירה. ניתנוו את מספר החילופים הצפוי תחת הנחה שסדר המועמדות הינו אקראי. אולם, מה נעשה אם לא ניתן להניח הנחה זו? למשל, אם אנו מקבלים את רשיימת המועמדות מחברת

כח אדם, אין לשלול את האפשרות שבחברת כח האדם ישנו גורם המUnoין להקשות עלינו ולຢיקר את תהליכי העסקה עבורה (אולי הוא מקבל עמלה על כל העסקה?), ויתכן והוא יסדר לנו את המועמדות בסדר רצוע בכונה. لكن, נרצה אלגוריתם שיבטיח את האקרואיות של הרשימה, כביכול "יפיל את הדפים על הרצפה ויסדר אותם מחדש". לצורך כך, נוסיף למודל החישובי שלנו רכיב של אקרואיות.

נניח כי נתונה פונקציה $Random(i, j)$ אשר בהינתן שני מספרים $\mathbb{N} \in i < j$ מחרירה מס' טבעי מהתחום $[j, i]$ בהתפלגות אחידה - כלומר הסתברות של $\frac{1}{j-i+1}$ לאיבר. נרצה להעזר בפונקציה זו לערבול של המערך באופן אחד לחולטי. נרצה לבחור פרמטריזציה אקרואית לחולטי.

הגדרה 3. בעיית בחירת תמורה:

קלט: המספרים $\{1, \dots, n\}$

פלט: תמורה אקרואית P של המספרים, כאשר הסיכוי לקבל כל אחת מהתמורות הוא בדיק $\frac{1}{n!}$.

10.9.1 שיטה ראשונה: הפלת הדפים אל הרצפה - ואיסופם מיין לשמאלי

אפשרויות אחת היא להגריל באקראי לכל איבר את המיקום החדש שלו. הבעיה היא שאם נגריל לכל איבר מספר בין 1 ל- n , יתכן בהחלתו שתיזכרנה התנשויות - שני איברים שנופלים באותו מקום. ואז לא כ"כ ברור כיצד נגדיר את התמורה (היא חד חד ערכית). אחת הדרכים להתגבר על הבעיה היא מראש להגריל את המספרים בטוחות גודל הרבה יותר:

אלגוריתם 1 ערבוב- A

$$A.length \rightarrow n . 1$$

2. יהיו $n..1$ מערך חדש

3. לכל i מ-1 עד n בצע:

$$\text{Random}(1, n^3) \rightarrow P[i] \text{ (א)}$$

4. מיין את איברי A תוך שימוש באיברי P המתאים כמפורט

כלומר הגרנו לכל איבר מספר בין 1 ל- n^3 , וסידרנו את האיברים לפי סדר ההגרלות. נרצה להשתכנע במספר דברים:

א. האלגוריתם פועל, ככלומר מוחזר תמורה כלשהי בהסתברות גבוהה מאוד.

ב. כאשר האלגוריתם מוחזר תמורה, הסיכוי לכל אחד מהפרמטריזציות הינו $\frac{1}{n!}$.

ג. זמן הריצה של האלגוריתם ייעיל.

למה 4. בהסתברות גבוהה של לפחות $\frac{1}{n}$ לא יהיה שני איברים $P[i] = P[j]$ כ"כ ש- $A_{i,j} = A_{j,i}$. נסמן ב- $A_{i,j}$ את המאורע שני איברים מסוימים מוקבים את אותו ערך. בהכרח כיוון שככל איבר מתקבל ע"י בחירות בתחום $[1, n^3]$ מותקיים $Pr(A_{i,j}) = \frac{1}{n^3}$. (הבחירה קורת זו אחר זו, בחרנו אחד וכך הסתברותו להיות 1 והסבירו שהhabva יהיה כמו זה $\frac{1}{n^3}$). אם כן, נסמן ב- $A = \bigcup_{i,j \in [n]} A_{i,j}$ את המאורע שקיים שני אינדקסים כלשהם שמקיימים את אותו ערך ב- P . מכאן לפי חסם האיחוד:

$$Pr(A) \leq \sum_{i,j \in [n]} Pr(A_{i,j}) = \binom{n}{2} \times \frac{1}{n^3} = \frac{n!}{(n-2)! \times 2 \times n^3} = \frac{n(n-1)}{2n^3} \leq \frac{n^2}{2n^3} = \frac{1}{2n} \leq \frac{1}{n}$$

ולכן, $Pr(\bar{A}) \geq 1 - \frac{1}{n}$.

למה 5. בהנחה שכל הערכים שהאלגוריתם הגריל שונים זה מזה, מוחארת תמורה אקראית בהתפלגות אחידה.

הוכחה: תהי $\pi = <\pi(1), \dots, \pi(n)>$ תמורה. נרצה להוכיח כי הסיכוי שהאלגוריתם יוציא את π הוא בדיק $\frac{1}{n!}$.

ההכללה לכל תמורה אחרת תשאוף דומה.

נסמן ב- A_1 את המאורע שהתא $P[1]$ קיבל את הערך הכי קטן במערך, ובאופן כללי לכל $n \leq i \leq n$ נסמן ב- A_i את הסיכוי ש- $P[i]$ ייריל את הערך ה- i בגודלו מבין כל ערכי המערך. אם כן, המאורע בו קיבלונו את $\pi = <1, \dots, n>$ הוא בדיק $\cap A_n \cap \dots \cap A_1$. אם כן,

$$Pr(A_1 \cap \dots \cap A_n) = Pr(A_1) \times Pr(A_2 | A_1) \times Pr(A_3 | A_1 \cap A_2) \times \dots \times Pr(A_n | \bigcap_{i=1}^{n-1} A_i) =$$

$$\frac{1}{n} \times \frac{1}{n-1} \times \frac{1}{n-2} \times \dots \times \frac{1}{1} = \frac{1}{n!}$$

עבור כל תמורה שאינה $\pi = <1, \dots, n>$ ההוכחה תעבור באופן דומה. באשר אם נסמן במאורע A_j את המאורע שתא $P[j]$ יכול ערך j והוכחה תהיה זהה בדיק על תמורה זההות.

כעת, ננתח את זמן ריצת האלגוריתם. בהנחה כי כל הגרלה מתבצעת ב- $O(1)$ זמן, שורות 3 – 1 מותבצעות ב- $O(n)$ זמן. אם נממש את המינון במינו בסעיף 4 נקבל שעលתו תהיה $O(n)$ וסה"כ אופטימלי של $O(n)$ זמן.

10.9.2 שיטה שנייה: "הווצאת האיברים ממקום סל בזיה אחריו זה,uschel hozachah nusheet baakrai bhetpulgot achida"

החסרון בשיטה הראשונה היא שלא בטוח שנתקבל תמורה, כיון שיתכן (בהתברות נמכה אמנים) שהאלגוריתם יגריל שני ערכים $[j, i] = P[i] = P[j]$. ניתן היה להציג מגנון שיתמוך עם תופעות כאלה בעזרת אקראיות נוספת, למשל הטלות מטבע שכיריעו בין איברים שב"תיקו" אולם צריך לוודא שבסקרה כזו אכן ההתפלגות אחידה למחרי. ו>Showן הריצעה לא ישפיע מדי. לכן, נרצה אלגוריתם שתמיד יעבד, בזמנו קבוע ללא תלות ב"מול" בהגדרות. כמובן, שההגדרות כו' יישמשו אותן – כדי להבטיח שהסיכוי לכל תמורה יהיה זהה. הרעיון הוא לחת את האיברים שלנו, לפזר אותם במקומות חדש של n תאים. בשלב הראשון, נצטרך לבחור מי יהיה האיבר הראשון. נבחר כל איבר בהסתברות $\frac{1}{n}$. לאחר מכן ישארו לנו $1 - n$ איברים ונבחר אחד מהם בהסתברות $\frac{1}{n-1}$. וכך הלאה, נבחן כי בכל שלב אחריו שבחרנו את i האיברים הראשונים נותרנו עם $i - n$ איברים שטרם שובצו. במקומות להקצות רישימה חדשה לתמורה והסופי, נינן לבצע את החישוב *place in place*. נוכל לשמר את התוכנה שבכל שלב i האיברים הראשונים הם כבר פיי סיידור הפלט $i - n$ האיברים האחרונים הם האיברים שטרם סודרו. כאשר האלגוריתם בוחר איבר למקומו i הוא מבצע החלפה בין האיבר שנמצא במקומו i לאיבר שצריך להיות שם באמצעות פועלות *swap*. להלן האלגוריתם:

ערובב אחידי-2 :
 $A.length \rightarrow n$
 א. $i \in [1, n]$ בצע:
 ב. $A[random(i, n)]$ עם $A[i]$

ברור, כי זמן ריצת האלגוריתם הינו $O(n)$. ההתברות לקבל תמורה כלשהי הינה $\frac{1}{n!}$ [במערך התרגול].

11 (טרגול מס' 11) Quick Select ו-Skip List

11.1 תיאור המבנה

רשימת דילוגים היא מבנה נתוני שנוצר ברנדומיות ואקראיות ותומך בשלוש הפעולות הבאות:
 א. הכנסה של איבר חדש.
 ב. חיפוש של איבר ע"פ המפתח שלו.
 ג. הוצאת איבר קיים.

כל הפעולות הללו נעשות בתוחלת זמן $O(\log n)$ או ב($O(\log n)$ זמן בהסתברות גבוהה). מיקום מבנה הנתונים הוא בתוחלת ($O(n)$).

מבנה הנתונים בניו מרומות אוטן נסמן L_0, L_1, L_2, \dots . כל רמה מכילה איברים, אשר שמורים כרשימה מקוישת ממויינית מהמספר הקטן ביותר עד המספר הגדול ביותר.

לצורך נוחות, בכל אחת מהרשימות ישנו המספרים $-\infty, \dots, \infty$, בשביל שתמיד יוכל לבצע השוואות להיוות בטוחים שהאיבר שאנו חפש לא נמצא בקצוות.

בכל רגע נתון הרמה התחתונה L_0 מכילה את כל האיברים שבמבנה, וכל רמה מכילה תת-קובוצה של איברים שנמצאים ברמה שמתחרתיה. כלומר $L_0 \subseteq L_1 \subseteq L_2 \subseteq \dots \subseteq L_3$. כל הרמות מחזיקות את אביריהם בצורה ממויינית, וכל איבר מכיל מצביעים לאיבר הבא והקודם בرمתו ועל העותק שלו ברמה מתחת (פרט לרמה L_0 שהרי אין מתחתיה). נתאר את הפעולות על המבנה:

א. הכנסה: נחפש את מקוםו של האיבר ברמה התחתונה (נתאר כיצד מחפשים מיד) ונכניס אותו למקום שהוא צריך להיות בו. לאחר מכן נטיל מטיבע: אם יצא עז - נוסיף עותק נוסף של המטיבע בrama מעלי. אם יצא פלי - נסימן. אם יצא עז בrama מעלי, שוב נחזיר על התהיליך ונטיל מטיבע: מתי נפסיק? בrama מסוימת שבה יצא פלי.

ב. חיפוש של איבר עס מפתח x: נתחל מהרמה העליונה בצד שמאל, בכל שלב נלקח ימינה בrama, עד שנגיע לאיבר הגדול ביותר בrama שקטן או שווה ל- x , אם הגיענו ל- x עצמו נרדף לעותק שלו מטה ב- L_0 באמצעות המצביעים וסימנו. אחרת: נרדף מהרמה אחת מטה, וממשיך ב巡视ה בrama מתחת מהאיבר אשר אליו הגיענו וכך באותו אופן. כאשר נגע לrama התחתונה L_0 - אם x מאוחסן במבנה הנתונים אז הוא בהכרח יופיע שם, אם לא נמצא אותו שם: סימן שאין מאוחסן במבנה.

ג. מחיקה: נחפש איבר, שנמצא נמחק אותו ואת כל עותקיו במבנה.

11.2 ניתוח סיבוכיות מקום וזמן ריצה של המבנה

עבור ערך x שהוכנס למבנה, נאמר כי גובהו של x הוא המספר הגדול ביותר i כך $x \in L_i$. כלומר,

$$h(x) = \max\{i | x \in L_i\}$$

 הגובה של המבנה בכל רגע נתון הינו: $h_{SL} = \max\{h(x) | x \in L_0\}$

משפט 1. לכל איבר x מתקיים כי $\mathbb{E}[h(x)] = 2$
הוכחה: הגובה של כל איבר הוא משתנה גאומטרי, עם הסתברות $\frac{1}{2}$ לשוווז"ל שלב הבא. ולכן

$$\mathbb{E}[h(x)] = \frac{1}{2} = 2$$

משפט 2. כהסתברות גובהה מתקיים כי $h_{SL} = O(\log n)$
הוכחה: האיבר מגע לגובה h ומעלה אמ"מ יצאו לו h הטלות וצופות של עז. הסיכוי לכך היו $\frac{1}{2^h}$
 (הטלות ב"ג). לכן, לפי חסס האיחוד:

$$\Pr[h_{SL} \geq h] = \Pr[\bigcup_{i \in [n]} h(x_i) \geq h] \leq \sum_{i \in [n]} \Pr[h(x_i) \geq n] = \frac{n}{2^h}$$

לכן, עכור האיבר מגע לגובה $h = c \log n$ נקבע כי $\Pr[h_{SL} \geq c \log n] = \frac{1}{2^{c \log n}} = \frac{1}{n^{c-1}}$, כלומר אקו $O(\log n)$ כהסתברות גובהה.

משפט 3. תוחלת גודל מכנה הנטויס (כמויות האיברים בכל הרופות יחד) היא $O(n)$
הוכחה: מס' האיברים ברמה התחתונה הוא n . תוחלת מס' האיברים בכל רמה הוא חצי מהרמה
 שמתתיה וכאן תוחלת מס' האיברים ברמה זו הוא $(\frac{n}{2})^i O$. סה"ג נקבל:

$$\mathbb{E}[|sizeSL|] = \mathbb{E}\left[\sum_{i=1}^{\infty} |L_i|\right] = \sum_{i=1}^{\infty} \frac{n}{2^i} = 2n = \Theta(n)$$

11.3 זמן פעולות חיפוש

על מנת לנתח את הזמן לפעולות חיפוש נסתכל על סדרת הפעולות בחיפוש בכיוון הפוך. נבחן כי ניתן לתאר את החיפוש כמורכב מפעולות ימינה בתוכה רמה, ותנועות למיטה מרמה אחת לאחרת. זמן הריצה הוא סך הפעולות, בתוספת פעולה אחת לכל היוטר לכל רמה, וכיוון שמספר הרמות הוא בהסתברות גבוהה $O(\log n)$ נותר לנו לחסום את אורך התנועות ימינה ולמטה.

נסתכל על סדרת הפעולות מהסוף להתחלה מהאיבר האחרון - האיבר אותו חפשנו. בכל צעד אנחנו נמצאים בקודקוד אחר במבנה, הצעד שלנו יהיה צעד לפני מעלה (נסמננו ב \uparrow) אם לקודקוד זה יש עותק ברמה מעלה, וצד שמאליה (נסמננו ב \leftarrow) אם לקודקוד זה אין עותק ברמה מעלה. כזכור, לכל קודקוד יש עותק ברמה מעלייו אמם בטלת המطبع שלו התקבל עץ - שיאהו מאורע בהסתברות חצי. נשים לב שהאלגוריתם מבצע צעד שמאליה רק אם יכול לבצע צעד לעילו, ככלומר אם בטלת המطبع של אותו הקודקוד התקבל פלי. כמות הצעדים למטה שאלגוריתם החיפוש מבצע חסומה כפובה בגובה המבנה. כדי לחסום את כמות הצעדים שמאליה שאלגוריתם החיפוש מבצע, מפסיק לחסום את כמות הפעמים שביקרנו בקודקוד שלא היה ניתן צעד למטה במהלך אלגוריתם החיפוש, וכן נרצה להוכיח את הלמה הבאה:

משפט 4. כסדרת הטילות כלות של מטבע הוגו, תוך $O(\log n)$ הטילות יתאפשרו לפחות $clogn$ הטלות של עץ, בהסתברות גבוהה.

הוכחה: נסתכל על סדרה של $10clogn$ הטילות. מחשב את הסיכוי שיש סדרה זו או פחות מ- $9clogn$ הטלות עץ. מאוועז זה זהה למאורע שישן לפחות $9clogn$ הטלות פלי. יש סה"ג $\binom{10clogn}{9clogn}$ חירות של מיקומות להטלת פלי. בהינתן חוויה של $9clogn$ מקרים, הסיכוי שככל המיקומות הללו הוטלו יהיו לפחות $2^{-9clogn}$. מכאו לפיו חפס האיזוח:

$$Pr[there \text{ are } less \text{ than } clogn \text{ tails}] = Pr[there \text{ are } at \text{ least } 9clogn \text{ heads}] \leq \binom{10clogn}{9clogn} \times \frac{1}{2^{9clogn}} = \binom{10clogn}{clogn} \times \frac{1}{2^{9clogn}}$$

לפי הנוסחה הכלאה: $\binom{n}{k} \leq (e \times \frac{n}{k})^k$

$$= \binom{10clogn}{clogn} \times \frac{1}{2^{9clogn}} \leq (e \times \frac{10clogn}{clogn})^{clogn} \times \frac{1}{2^{9clogn}} = (10e)^{clogn} \times \frac{1}{2^{9clogn}}$$

$$= (\frac{10e}{2^9})^{clogn} < (\frac{1}{2})^{clogn} = \frac{1}{n^c}$$

ולכן, הסיכוי שיש לפחות $clogn$ הטלות עץ הינו $\leq -\frac{1}{n^c}$, אך סיכוי גבוה.

כעת, נזיר בлемה 4 לחסימת זמן החיפוש בתוחלת.
 נסמן ב- A את המאורע כי גובה המבנה הוא לכל היוטר $clogn$. נסמן ב- B את המאורע שיתבצעו לפחות $clogn$ צעים \uparrow בתוקן סדרה של עד $10clogn$ צעים. נראה כי:

$$Pr[\overline{A \wedge B}] = Pr[\overline{A} \vee \overline{B}] \leq Pr[\overline{A}] + Pr[\overline{B}] \leq \frac{1}{n^{c-1}} + \frac{1}{n^c} \leq \frac{2}{n^{c-1}} \leq \frac{1}{n^{c-2}}$$

כלומר, ההסתברות למאורע זה היא $\Omega(n^{-c})$. מה מאויע זה אומר? ההסתברות שוגבה המבנה הוא יותר $c \log n$ (נעה הרבה מעלה) או לחופין שיתבצעו פחות מ- $c \log n$ פעמים (יטילו פחות מדי עץ מה שאומר שהרבה פלי תזוזה שמאליה) - היא חסומה ב- $\frac{1}{n^{c-2}}$. ولكن ההסתברות שפעולות החיפוש תצליח בזמן לוגריטמי די גבוהה $\leq \frac{1}{n^{c-2}} \cdot \Omega(n^{-c}) = \Omega(n^{-2c+2})$.

פעולות הכנסה תמיד מתחילה בפעולות חישוב, כדי למצוא את מקומו האיבר המוכנס בرمמה התחרתונה. לאחר החיפוש נדרש ליצור עותקים נוספים ע"פ הטלות המetu. כיוון שתוחלת מס' הרמות שיתווסף היא $O(\log n)$, רמות וכל הוספה לרמה לוקחת $O(1)$ זמן איזה סה"כ הכנסה לא לוקחת יותר מהחיפוש וכל בנסיבות גובהה גבוהה $O(\log n)$ גם כן. עברו המהיקה - בדומה. (מוחלטים בחיפוש ומוחקים מהרמות השונות).

Quick Select 11.4

אלגוריתם QS מקבל מערך $A[1, \dots, n]$ של n מספרים שונים ומספר טבעי k ($1 \leq k \leq n$) ומציין את האיבר k בגודלו במערך.

הפונקציה $i = Part(A, i)$ מקבלת כקלט מערך $A[1, \dots, n]$ של n מספרים שונים ואינדקס $1 \leq i \leq n$. נסמן $x = A[i]$. יהיו p האינדקס של x בסדר הממוין של A . הפונקציה מסדרת את A מחדש כך x יהיה במקום p , כל האיברים שקטנים מ- x ימצאו בתת המערך $A[1, \dots, p-1]$ וכל האיברים הגדולים מ- x יהיו בתת המערך $A[p+1, \dots, n]$. כמו כן, מחזירה את p כפלט. להלן האלגוריתם:

אלגוריתם 1	$QS(A[1..n], k)$
<hr/>	
$Part(A, rand(1, n)) \rightarrow p$	

1. אם $p = k$

2. אם $p > k$

3. אם $p < k$

4. אחרת, החזר $QS(A[p+1..n], k-p)$

האלגוריתם בוחר בכל קריאה רקורסיבית *pivot* אקראי ומבצע עלי' חלוקה, וממשיך רקורסיבית בכיוון שבו נמצא האיבר שהתקבל בקלט. נסמן ב- n את גודל תת המערך של A שנותך בקלט בرمמה *how* של הרקורסיבי.

משפט 5. מתקיים $Pr[n_{i+1} > \frac{3}{4}n_i] \leq \frac{1}{2}$ הוכחה: נניח כי בנסיבות לפחות חצי למשהו האינדקס p של *pivot* יהיה בטוח $\frac{3}{4}n_i \leq p \leq \frac{3}{4}n_{i+1}$ נמוך ש-*זאת*, אך ככלו גזול לפחות $\frac{n}{4}$ ימוך מהמעו. ככלו הוכחו כי $\Pr[n_{i+1} \leq \frac{3}{4}n_i] \geq \frac{1}{2}$ והוא מוכיח ש-*זאת*.

נסמן ב- X את זמן היריצה עד לסיום האלגוריתם. נרצה לחשב את תוחלת זמן היריצה. נאמר כי קריאה רקורסיבית i הינה טובה אם מתקיים $n_{i+1} \leq \frac{3}{4}n_i$. ככלומר: לאחר ביצועה נפטרנו לפחות מרבע מגודל הקלט הקודם. נסמן b את מס' הקרייאות הטובות, נבחן כי לאחר כל קריאה צו' גודל תת המערך קטן פי $\frac{3}{4}$. נרצה לבדוק כמה איטרציות טובות ישנו לכל היותר. ככלומר, אם $m > k$ אז $(\frac{3}{4})^k < 1$. נסמן את מס' הקרייאות הטובות ונסמן ב- X_j את זמן היריצה של האלגוריתם מתחילה בקריאה הטובה ה- j עד הקריאה הטובה ה- $j+1$. בפרק זמן זה יתכו נमובן קרייאות רעות. נסמן ב- D_j את מס' הקרייאות הרעות בין הקריאה הטובה ה- j לבין ה- $j+1$. נסמן n כמספרם.

הצלחה לפחות $\frac{1}{2}$ ולכון $2 \leq \mathbb{E}[D_j]$. אם כן, נוכל לחסום את ע"י מס' הקריאה והרעות שמתבצעות כפולה זמן לקריאה הגדולה מבניהן. סה"כ נקבל $n \times (\frac{3}{4})^j \leq D_j$. מכאן:

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_j X_j\right] = \sum_j \mathbb{E}[X_j] \leq \sum_j \mathbb{E}[D_j] \times n \times (\frac{3}{4})^j \leq 2n \sum_j (\frac{3}{4})^j = O(n)$$

חסם עליון על זמן הריצה בהסתברות גבוהה

נוכיה כי חסם עליון על זמן הריצה הינו $O(n \log n)$ בהסתברות גבוהה של לפחות $1 - \frac{1}{n^c}$ עבור $c > 0$. בתחילת, נבחן כי מס' האיברים שנותרו לאחר כל קריאה רקורסיבית הינו:

$$\mathbb{E}[n_{i+1}] \leq \frac{1}{2} \times \frac{3}{4} n_i + \frac{1}{2} \times n_i = \frac{7n_i}{8}$$

נרצה לנתח באופן מדויק כמה איברים נמחקים בכל שלב. נבחן כי הסיכוי של האיבר ה i בגודלו להמחק מהמערך הוא $\frac{|r-k|}{n}$. כאשר k הוא האינדקס של האיבר אותו אנו מחפשים. סמן b_r במשתנה אינדיקטורי עבור המאوروע כי האיבר ה i בגודלו נמחק מהמערך. נקבל כי תוחלת מס' האיברים הנמחקים לאחר הسابוב i הינו:

$$\mathbb{E}\left[\sum_{r=1}^k X_r + \sum_{r=k+1}^{n_i} X_r\right] = \sum_{r=1}^k \frac{k-r}{n_i} + \sum_{r=k+1}^{n_i} \frac{r-k}{n_i} =$$

$$\dots \geq \frac{n_i}{4}$$

כלומר, תוחלת מס' האיברים שנמחקו הוא לפחות רביע מגודל תות המערך שהתקבל בקלט. לכן, תוחלת מס' האיברים שנשארו הוא לפחות $\frac{3}{4}n_i$. נוכיה כי בהסתברות לפחות $1 - \frac{1}{n^c}$ ישן $O(\log n)$ קריאות רקורסיביות בריצת האלגוריתם עד לסיוםו. נעזר dabei שוויון מרקוב. נראה כי בהסתברות גבוהה מתקיים המאורע:

$$Pr[n_{c \log_{\frac{4}{3}} n} < 1]$$

עבור c קבוע. אם כן, נציב $n = (c+1) \log_{\frac{4}{3}} n$ ונקבל:

$$\mathbb{E}[n_{(c+1) \log_{\frac{4}{3}} n}] \leq \left(\frac{3}{4}\right)^{(c+1) \log_{\frac{4}{3}} n} \times n = \frac{1}{n^c}$$

ולכן לפי אי שוויון מרקוב:

$$Pr[n_{(c+1) \log_{\frac{4}{3}} n} > 1] \leq \frac{\mathbb{E}[n_{(c+1) \log_{\frac{4}{3}} n}]}{1} = \frac{1}{n^c}$$

ולכן, בסיכוי של לפחות $1 - \frac{1}{n^c}$ לאחר $(c+1) \log_{\frac{4}{3}} n = O(\log n)$ קריאות רקורסיביות נשארנו עם פחות מאייר אחד. עלות זמן הריצה של כל קריאה רקורסיבית הינה $O(n_i)$ ולכן בסיכוי גבוה זמן הריצה של האלגוריתם כולל הינו $O(n \log n)$.

12 הרצתה 11: Chernoff חסמי

בהרצתה זו נלמד כלי הסתברותי מתמטי, ובהמשך נדון כיצד למשר קלים אלו באלגוריתמים.

12.1 אי שוויון מרקוב וצ'בישב

נניח כי אנחנו מטילים מטבע מסוימת של פעמים ונרצה להבטיח כי בסיכוי גבוה (של לפחות $\frac{1}{n^c} - 1$) נ בטיח שיצא עז לפחות פעם אחת. הסיכוי שלא יצא עז אף פעם הוא $\frac{1}{2^k}$. נראה כי אם $k = clog(n)$ אז בנסיבות לפחות $\frac{1}{n^c} - 1$ נקבל עז.

מה אם נרצה $\log(n)$ פעמים שיצא עז בסיכוי גבוה פולינומי? נתיל (n) פעמים, בכל הטלות הסיכוי לקבלה עז הוא $\frac{1}{n^c} - 1$. הסיכוי שקיימות סדרה של לפחות $M(\log^2(n))$ הטלות שבה לא יצא עז $\geq \frac{\log(n)}{n^c} > \frac{1}{n^{c-1}}$. נבחן כי במחשבה זו מוגדים $L(\log^2(n))$ הטלות. צ'רנוフ ראה שמספריים $\Theta(\log n)$ הטלות - זו החזקה של צ'רנוフ שנרצה להשתמש בה.

אי שוויון מרקוב: אם X מ"מ אי שלילי, וכי $Pr[X > k] < \frac{\mathbb{E}[X]}{k}$ ויהי $0 < k \leq \mathbb{E}[X]$.
 $Pr[X > k\mu] < \frac{1}{k}$.

אי שוויון צ'בישב: אם X מ"מ (לא בהכרח אי שלילי), ונסמן $\mu = \mathbb{E}[X]$ וכן $\sigma^2 = Var[X]$ נבחן כי הטעות b היא ליניארית במכנה של ההסתברות. בפרט, $\mathbb{E}[(X - \mu)^2] \geq t^2$.

$$Pr[|X - \mu| \geq t] \leq \frac{\sigma^2}{t^2}$$

$Pr[X > k\mu] = Pr[X - \mu > (k - 1)\mu] <_* Pr[|X - \mu| \geq (k - 1)\mu] \leq \frac{\sigma^2}{(k - 1)^2 \mu^2}$

שכן * נכו כי בהוספת ערך מוחלט נוסף הסתברויות (אפשרויות לערכים) שלא היו קודם, מס' שחיי שליליים קודם כעת נכנים.

כעת נבחן, כי הקשר בין $(k - 1)\mu$ לבין t (במכנה הוא כעת ריבועי). יותר טוב ממרקוב. (כל עוד σ, μ מתנהגים "יפה" - אם לא יתנהגו יפה כלומר לא יהיו קורובים יחסית זה זהה, יתכן שעדיין להשתמש במרקוב).

ההוכחה של צ'בישב:

$$Pr[|X - \mu| > t] = Pr[(X - \mu)^2 > t^2] \leq \frac{\mathbb{E}[(X - \mu)^2]}{t^2} = \frac{Var[X]}{t^2} = \frac{\sigma^2}{t^2}$$

כעת נבחן כי לא חייבים להעלות רק בריבוע, אלא כל העלה של פונקציה **מוניוטונית** שתהפוך את המשתנה לאי שלילי בהכרח (לכן העלה בשלישית לא הייתה עובדת).

טענה: ידי X מ"מ $\mathbb{E}[X] = \mu$, איז $t > 0$ ולכל $1 \leq \ell \leq 2\ell - moment$:

$$Pr[|X - \mu| > t] < \frac{\mathbb{E}[(X - \mu)^{2\ell}]}{t^{2\ell}}$$

הוכחה:

$$Pr[|X - \mu| > t] = Pr[(X - \mu)^{2\ell} > t^{2\ell}] \leq \frac{\mathbb{E}[(X - \mu)^{2\ell}]}{t^{2\ell}}$$

נניח כי הטענו מטבע $8\log(n)$ פעמים. מה הסיכוי שייצאו לפחות $\log(n)$ פעמים לפחות. נסמן ב- X את מס' הפעמים שייצא עז.

לפי אי שוויון מרקוב: נعزيز במאורע המשלבים (יהו $X - 8\log(n)$ הטענו $8\log(n)$ הטענו של פלי, נראה כי הסיכוי לכך שייצאו לכל היותר $\log(n)$ הטענו של עז (המשלים) שווה לסיכוי שייצאו לפחות $7\log(n)$ פעמים (פלוי)

$$Pr[8\log(n) - X > 7\log(n)] < \frac{\mathbb{E}[8\log(n) - X]}{7\log(n)} = \frac{8\log(n) - \mathbb{E}[X]}{7\log(n)} =$$

נבחן $\mathbb{E}[X] = \frac{1}{2} \times 8\log(n) = 4\log(n)$.

$$= \frac{8\log(n) - 4\log(n)}{7\log(n)} = \frac{4}{7}$$

לפי אי שוויון צ'בישב:

$$Pr[8\log(n) - X > 7\log(n)] = Pr[(8\log(n) - X) - 4\log(n) > 3\log(n)]$$

$$\leq Pr[|4\log(n) - X| > 3\log(n)] = Pr[|X - 4\log(n)| > 3\log(n)] \leq_* \frac{2\log(n)}{(3\log(n))^2} = \frac{2}{9\log(n)}$$

שכן * נכון כי נבחן שהשונות הינה $\frac{1}{2} \times 8\log(n) = 2\log(n)$ כעת, עוד לא פולינומי אך השתperfנו.

לפי 2ℓ – moment נקבל:

$$Pr[8\log(n) - X > 7\log(n)] \leq Pr[|4\log(n) - X| > 3\log(n)] <$$

$$< \frac{\mathbb{E}[(4\log(n) - X)^{2\ell}]}{(3\log(n))^{2\ell}}$$

ambil לפתוח את המתמטיקה (לא פתרנו זאת בהרצאה), נבחן כי נקבל כאן לבסוף לכל ℓ ממשו מהצורה של $\frac{1}{(\log n)^\ell}$ (בתוספת קבועים). כמובן:

$$\frac{\mathbb{E}[(4\log(n) - X)^{2\ell}]}{(3\log n)^{2\ell}} \approx \frac{1}{c \times \log^\ell(n)}$$

נבחרו: $\ell = \log_{\log(n)}(n) = \frac{\log(n)}{\log(\log(n))}$
בצ'רנוֹף כמובן.

12.2 חסמי צ'רנוֹף: הגדרה

הנחות יסוד: נניח כי X הוא סכום של משתנים מקרים אינדיקטוריים (כל אחד מהם יכול להתפלג שונה!) בלתי תלויים (מאוד חשוב!).
נסמן:

$$X = \sum_i x_i$$

הרעיון מבוסס על השיטה שבה עבדנו קודם. השתמשנו בהוכחות קודם בפונקציה מונוטונית אי-שלילית. איזו פונקציה מונוטונית אי-שלילית אנחנו מכירים? $e^x = f(x)$. בואו ננסה

$$Pr[X \geq a] =_{\forall t > 0} Pr[e^{tX} \geq e^{ta}] \leq_{markov} \frac{\mathbb{E}[e^{tX}]}{e^{ta}}$$

נרצה לחשב את $\mathbb{E}[e^{tX}]$.

$$\mathbb{E}[e^{tX}] = \mathbb{E}[e^{t \sum x_i}] = \mathbb{E}[\prod_i e^{tx_i}] =_* \prod_i \mathbb{E}[e^{tx_i}]$$

המעבר ה* חוקי כיון שהמשתנים המקרים ב"ת ולכן גם e^{tx_i} .

אי שוויון צ'רנוֹף: עבור $1 < \delta < 0$ מתקיים: $Pr[X > (1 + \delta)\mu] \leq \left(\frac{e^\delta}{(1 + \delta)^{1 + \delta}}\right)^\mu$
כמובן שזה נראה רע ומפיחיד, ולכן ישנן שתי גרסאות פשוטות יותר:
.1

$$Pr[X > (1 + \delta)\mu] < e^{-\frac{\mu\delta^2}{3}}$$

.2

$$Pr[X < (1 - \delta)\mu] < e^{-\frac{\mu\delta^2}{2}}$$

12.3 הוכחת אי שוויון צ'רנוフ

הערה. צבי אמר לנו לא נדרשים לדעת את ההוכחה בע"פ, אך העקרונות שלה כן חשובים ולכון היא גם מוגנת כאן.

באשר $\mu = (1 + \delta) \cdot a$ (ממה שפיתחנו קודם) נקבל:

$$Pr[X \geq (1 + \delta)\mu] \leq \frac{\mathbb{E}[e^{tX}]}{e^{t(1+\delta)\mu}}$$

נרצה לחשב את $\mathbb{E}[e^{tx_i}]$. כמובן נחשב את $\mathbb{E}[e^{tX}] = \prod \mathbb{E}[e^{tx_i}]$ נסמן: (נבחן הפונקציות לא בהכרח שוות הסתברות)

$$X_i = \begin{Bmatrix} 1 & p_i \\ 0 & 1 - p_i \end{Bmatrix}$$

$$\mathbb{E}[e^{tx_i}] = p_i e^t + (1 - p_i)e^0 = 1 + p_i(e^t - 1) \leq_* e^{p_i(e^t - 1)}$$

כאן * נעזרנו באי השוויון $1 + x \leq e^x$ לכז:

$$\mathbb{E}[e^{tX}] = \prod \mathbb{E}[e^{tx_i}] \leq \prod e^{p_i(e^t - 1)} = e^{(e^t - 1) \times \sum p_i} =_{**} e^{(e^t - 1) \times \mu}$$

** שכן $\mathbb{E}[x_i] = p_i$ ומלינאריות התוחלת: $\mathbb{E}[X] = \sum \mathbb{E}[x_i] = \sum p_i$
cut נחזור לאי השוויון מעלה:

$$Pr[X \geq (1 + \delta)\mu] \leq \frac{\mathbb{E}[e^{tX}]}{e^{t(1+\delta)\mu}} \leq \frac{e^{(e^t - 1) \times \mu}}{e^{t(1+\delta)\mu}} = (e^{(e^t - 1) - t(1+\delta)})^\mu$$

נבחר t על מנת להקטין את ההסתברות. נגדיר: $\phi(t) = (e^t - 1) - t(1 + \delta)$

$$\phi'(t) = e^t - (1 + \delta) = 0 \implies t = \ln(1 + \delta)$$

זה אכן מינימום ע"י גירה נוספת. cut:

$$e^{\mu(e^{\ln(1+\delta)} - 1 - \ln(1+\delta)(1+\delta))} =_*$$

$$(\frac{e^{1+\delta-1}}{e^{(1+\delta)\ln(1+\delta)}})^\mu = (\frac{e^\delta}{(1+\delta)^{1+\delta}})^\mu$$

$$* \text{ שכנ} e^{ln(1+\delta)} = 1 + \delta. \text{ כנדרש.}$$

כעת נחזר לדוגמה מקודם. $\mathbb{E}[Y] = 4\log(n)$. $Y = 8\log(n) - X$ כפי שראינו. נרצה לחשב חסם ל:

$$Pr[Y > 7\log(n)] = Pr[Y > (1 + \frac{3}{4})\mathbb{E}[Y]] < e^{\frac{-\mu\delta^2}{3}} = e^{\frac{-4\log(n)(\frac{3}{4})^2}{3}} = e^{-\frac{3}{4}\log(n)} = \frac{1}{n^\alpha}$$

עבור $\alpha > 1$ קבוע.

קבועים: נתבונן כי לבסוף עברנו אל $\frac{1}{n^\alpha}$ אך עליינו לוודה זאת (!) שכן $\alpha > 1$ אם זה לא היה עובד עם $8\log(n)$ ננסה $10\log(n)$ ואם לא מספיק אז ננסה $20\log(n)$ זה עדין O של. ב מבחנים עליינו בכך לוודה זאת שכן $\alpha > 1$.

מסקנה: עבור $\log(n)$ הטלות עצם מספיק להטיל $8\log(n)$ פעמים.

מסקנה: בעוד מרכיב נתן תלות ליניארית, צ'רנוフ נתן תלות לוגריתמית (ובהסתברות קבועה של לפחות $1 - \frac{1}{n^\alpha}$). כעת נקבל עם צ'רנוフ:

$$Pr[X > (k-1)\mu] \leq \left(\frac{e^{k-1}}{(k-1)^{k-1}}\right)^\mu$$

שהרי זו כבר תלות בחזקת $1 - k$.

12.4 מיון מהיר (Quick Sort)

ניתנו בהרצאה 10 את תוחלת מס' ההשוואות וראינו כי תוחלת מס' ההשוואות = זמן הריצה היא $O(n\log n)$. כעת, ננתח את החסם עליון על זמן הריצה בסיסי גובה.

נתבונן בעץ הרקורסיה של המיון. בתחילת בידינו n איברים שנכנסו אל עצם הרקורסיה. כל קודקוד, בחר מסלול מסוימ.

נתבונן בעלה בעץ הרקורסיה. נסמנו $a \in A$. מה הסיכוי שהעומק d_a של a הוא "גדול"? (גדול ממש מ $10\ln(n)$). נוכיח בשלב ראשון שהסיכוי לכך קטן פולינומית.

אייטרציה טוביה היא אייטרציה שבה כל קריאה רקורסיבית היא מגודל לכל יותר $\frac{3}{4}$ מגודל הקלט המקומי. הסיכוי לאיטרציה טוביה הוא בדיק $\frac{1}{2}$. כמה אייטרציות טובות ישנו? נבחן כי אם מס' האיטרציות הטובות $< \log_{\frac{4}{3}}(n)$ או בהפך נשארו 0 איברים.

מה ההסתברות שב $10\ln(n)$ אייטרציות שהבן a משתתף, לא היו (n) $\log_{\frac{4}{3}}(n)$ אייטרציות טובות? [שכנן שזה צעק: צ'רנוフ].

תוחלת מס' האיטרציות הטובות הינו $\frac{1}{2} \times 10\ln(n) = 5\ln(n)$. נחשב את הסיכוי שאנחנו רוחקים

מהתוחלת. נסמן X כמשתנה מקרי של מס' האיטרציות הטובות (שהוא סכום של משתני אינדייקטור ב"ת שכן כל איטרציה ב"ת באיטרציה אחרת) עבור עלה כלשהו, נרצה לחשב את:

$$Pr[X < \log_{\frac{4}{3}}(n)]$$

כלומר: ההסתברות כי מס' האיטרציות הטובות היה קטן מערך זה. נרצה לחלץ את δ . כיצד? נזכר בחוק הלוגריתמי לשוני בסיס:

$$\log_b(a) = \frac{\ln(a)}{\ln(b)}$$

$$\log_{\frac{4}{3}}(n) = (1 - \delta) \times 5\ln(n) \implies 1 - \delta = \frac{\log_{\frac{4}{3}}(n)}{5\ln(n)} = \frac{\frac{\ln(n)}{\ln(\frac{4}{3})}}{5\ln(n)} = \frac{1}{5\ln(\frac{4}{3})}$$

$$\implies \delta = 1 - \frac{1}{5\ln(\frac{4}{3})} \approx 0.3$$

נבחן כי אכן $0 < \delta < 1$ וזה מותאים לתנאי צ'רנוף. לכן:

$$Pr[X < \log_{\frac{4}{3}}(n)] = Pr[X < (1 - \delta)\mu] \leq e^{-\frac{\mu\delta^2}{2}}$$

$$= e^{-\frac{5\ln(n)\delta^2}{2}} = e^{\ln(n)^{-\frac{5\delta^2}{2}}} = n^{-\frac{5\delta^2}{2}} = \frac{1}{n^{\frac{5\delta^2}{2}}} = \frac{1}{n^\alpha}$$

עבור $1 > \alpha$ (הערה, בדוגמה שלקחנו זה לא קורה. היה צריך לבחור $20\ln(n)$ למשל. בכל מקרה מדובר ב- O של).

מכאן, שבסיסי גבוה של לפחות $\frac{1}{n^\alpha} - 1$ העומק של האיבר הנ"ל בעז הרקורסיה הוא לוגרייטמי (יהיו $\log_{\frac{4}{3}}(n)$ איטרציות טובות ונסיים אליו).

עתה נשאל מהו הסיכוי שקיים עלה עמוק כלשהו, מה הוכחנו עבור איבר ספציפי? הראיינו כי עבור $a \in A$ כלשהו מתקיים:

$$Pr[depth(a) > 10\ln(n)] \leq \frac{1}{n^\alpha}$$

כלומר: ההסתברות להתקע בעומק גדול מדי די קטנה. מכאן עבור כל האיברים - יש לכל היותר n איברים (עלים פוטנציאליים). נרצה לדעת מה ההסתברות שקיים לפחות איבר אחד שהענף שלו ארוך מדי. כלומר:

$$Pr[\exists a : depth(a) > 10\ln(n)] \leq \sum_{i=1}^n Pr[depth(a_i) > 10\ln(n)] \leq \frac{n}{n^\alpha} = \frac{1}{n^{\alpha-1}}$$

ולכן,

$$Pr[\forall a : depth(a) < 10\ln(n)] = \overline{Pr[\exists a : depth(a) > 10\ln(n)]} \geq 1 - \frac{1}{n^{\alpha-1}}$$

ישנה תלות בין האיטרציות. לכן בשלב זה השתמשנו בחסם האיחוד ולא בצרנוף. ככלומר: ההסתברות של העץ יהיה בעומק לוגריטמי היא לפחות $\frac{1}{n^{\alpha-1}} - 1$ עבור $\alpha > 1$ קבוע כלשהו. כאשר עומק העץ הוא $O(\log n)$, בכל רמה מתבצעת לכל היותר $O(n)$ חישובות - חישובות (partition) איזי זמן הריצה הכלול של האלגוריתם (בהסתברות גבוהה) יהיה:

$$O(n) \times O(\log n) = O(n \log n)$$

מה הוכחנו כאן? זמן הריצה בוריסט קיס - הוא $O(n \log n)$ - בסיסי גובה.

12.5 ניסוח שונה לא שיוויון צ'רנוף

$$Pr(|X - \mu| \geq \delta\mu) \leq 2e^{-\frac{\delta^2\mu}{3}}$$

13 הרצתה 12: MST מתקדם

הערה. הרצתה לא תהיה פורמלית וכן יהיו טענות רבות שייאמרו ללא הוכחה.

13.1 מבוא

גנזה לאורך כל הרצתה: קיים יחס סדר מלא על הקשתות, נקרא לפונקציה זו $ID(e)$ לכל קשת $e \in E$. בהינתן שיש יחס סדר מלא על הקשתות ניתן לומר $w(e), Id(e) <_{Lex} e < e'$ אם $w(e), Id(e) <_{Lex} e'$ (מסתכמים ראשית על המשקל, אם הקשתות הן באותו משקל אז ישנה דרך לשבור סימטריה לפי Id). מסקנה: כל הקשתות במשקלים (לפי יחס הסדר) שונות).

מסקנה: הרצתה של קروسקל לפי יחס סדר זה תניב תמיד עץ פורש מינימום יחיד (שכן תמיד ישנו אותו סדר למינון בשל שבירת הסימטריה).

זכור כי ראיינו שתי תכונות עיקריות:

1. קשת קלה בחותך \iff קיים MST שמכליל את e .
2. קשת כבדה במעגל \iff קיים MST שלא מכיל את e .

בקשר של יחס סדר המלא: קשת e שקופה בחותך (עם id הכי קטן) תהיה בכל MST קשת e שביחס סדר מגיעה הכי מאוחר (עם id הכי גדול) לא תהיה באף MST.

Boruvka 13.2

מדובר באלגוריתם הראשון שהומצא ל-MST.

סביר את הרעיון ללא פסאודו קוד:

מתחילה מוסף של קודקודים של הגרף, ישנן קשתות בגרף אך לא מציירים אותן עדין - כל קודקוד בוחר את הקשתה הקטנה ביותר שלו לפי יחס הסדר. נבחן כי ישנים שני מקרים:

אם עבר קודקוד v הקשת הכי קטנה היא (u, v) והקשת הכי קטנה עבורה u היא גם (u, v) אז מעולה. אך זה לא בהכרח - יתכן שלא יש קשת טובה יותר טובה.
נבחון: בהכרח לא יוצר מעגל! בשלילה שיש מעגל איזי יש קודקוד ש-2 קשותות חלות בו, ונקבל סתייה ליחס הסדר במעגל (יהיה קודקוד שבחר בקשת הלא טובה ביותר שלו).

טענה: בעקבות יחס הסדר, הקשותות שנבחרו בסיום התהליך לא יצרו מעגל.

נוסף את כל הקשותות LST , נסמן T .
בשלב הבא, נכווץ את הקשותות שנבחרו (ננסה לדמיין את התהליך ככה שבתחילת כל קודקוד הוא רכיב קשירות בפני עצמו. אם קודקוד 1 וקודקוד 2 חוברו ביניהםCut המחשבים לקודקוד אחד בלבד 12 - קודקוד גדול!).
נבחן כי אם קודם הוי $|V|$ קודוקדים, לאחר הכווץ יהיה לכל היותר $\frac{|V|}{2}$ קודוקדים (המקרה הכי טוב שככל שניים בחרו אותה קשת)
נמשיך ברקורסיה עד למקרה הבסיס (קודקוד אחד) - יהיה לנו סה"כ $O(\log|V|)$ איטרציות.

מעט יותר פורמלית:

בכל איטרציה:

- א. כל הקודוקדים בגרף הם מבודדים (או מגה הקודוקדים שיצרנו)
- ב. לכל קודקוד $V \in T$ נמצא את הקשת e_v המינימלית המוחברת אליו.
- ג. הוסף את כל הקשותות e_v שמצאת אל MST
- ד. כיווץ: אגד את כל הקודוקדים שחויבו ע"י הקשותות הללו ל"קודוקדים גדולים" ומחק לו לאות עצמיות (קשותות שחברו בין קודוקדים שכעת באוטו קודקוד) - הקשותות שיישארו לך בגרף הם רק קשותות שחברו בין גושים של קודוקדים גדולים.

נבחן כי علينا למצוא דרך להפוך מהלולאות העצמיות - בכל כיווץ נפעיל DFS למצוא את הרכיבים הקשיים ואם קשת שנייה קצוטה הוי באותו רכיב מדובר Cut בולאה העצמית שנפטר ממנו. זה לא יגע בזמן הריצה אסימפטוטית. (הערה: המימוש הטכני תלוי לפרשנות וישנם דרכים רבות. למשל: בעת שהוא מאחד את הקודוקדים הוא נותן ID חדש לכל אחד מהם, ובסיום הבא שירץ DFS ויראה קשת שנייה הקודוקדים בה מרכיב עם אותו ID הוא יתעלם מהקשת והוא יカリ עליה בולאה עצמית).

זמן הריצה:

פרק את הנитוח לשניים:

1. כמה עולה סבב?

* בכל סבב כל קודקוד עובר על כל קשותותיו ומוצא את הקשת המינימלית. לכן $= O(\sum_{v \in V} \deg(v)) = O(|E|)$
* בשביל לדעתוizia קודקוד התאחד עם מי, מרכיבים DFS\BFS על הקשותות שנבחרנו + עוברים על הקשותות ומוחקים את אלו שהפכו לlolאות עצמיות. סה"כ $O(|V| + |E|) = O(|E|)$
לכן סבב עולתו $O(|E|)$

2. כמה סבבים ישנים?

נסמן V_i את מס' הקודוקדים בשלב ה- i . נבחן כי בהכרח $|V_i| \leq \frac{1}{2}|V_{i-1}|$ (במקרה הגרוע ביותר: כל זוג קודוקדים מתחדים לקודקוד אחד) ולכן $\log|V|$ האיטרציות הינו $O(\log|V|)$ לכן סה"כ זמן הריצה הוא $O(|E|\log|V|)$.

13.3 שיפור Yao

יאו יש ובאמת: בורובקה עשה רעיון יפה - אך למה שלכל קודקוד נטרוק את כל הקשותות שיצאוות ממנו? אם אנחנו יודעים כי קשת היא מאוד יקרה - אולי לא כדאי להסתכל עליה בסיבובים הראשונים.

יאו הסתכל על קודקוד $V \in u$ ועל כל הקשתות שיצאו ממנו (ישנן $\deg(u)$ קשתות כאלה) וחילק את קשתות u ל k קבוצות (*Buckets*): נסמן $E_u^{(1)}, E_u^{(2)}, \dots, E_u^{(k)}$ ועבור כל קבוצה שכזו מתקיים כי גודלה הוא $\lceil \frac{\deg(u)}{k} \rceil$.
יחס הסדר צריך להיות כך כי כל מי שבקבוצה ראשונה קטן ממי שבקבוצה השנייה, וכי שבשניתה, קטן מהשלישית וכו'.

כיצד יוצרים את החלוקה הזו? נניח כי k חזקה של 2, ונשתמש באלגוריתם *Select* למציאת החציון. מכאן נקבל שתי קבוצות (גדולים וקטנים מהחציון), ונמשיך ברקורסיה על כל צד (של אלו שגדולים מהחציון ואלו שקטנים ממנו). כך נמשיך ברקורסיה עד שנקבל k קבוצות בדיק. למעשה, כיוון שבכל שלב אנחנו מחלקים את הקבוצות ל-2, מס' הפעמים שנפטרך לבצע את הרקורסיה הוא $O(k \log k)$. כמה עולה כל הרצאה? אלגוריתם סלקט עולה זמן לינארי. כאן הלינאריות הינה $O(\deg(u) \log k)$ - רק הקשתות של אותו קודקוד. לכן סה"כ זמן הריצה $O(\deg(u) \times \log(k))$.

כעת, קודקוד $V \in u$ צריך לחפש את הקשת הולמתה ביותר שיצאת ממנו. הוא לא צריך (בזיותה) לבדוק העזר המקדימה) לעבור על כל השכנים שלו. אלא: רק על $E_u^{(1)}$. לכל קשת הוא יבודק: האם הקשת מחברת אותו לקודקוד שנמצא אליו? אם כן - מדובר בולאה עצמית. האלגוריתם יזרוק את הקשת הזו וימשיך לקשת הבאה באותו חיבור. נבחין: הוא עירף אותה לתמיד.
אם לא - מצאנו את הקשת הולמתה. זו הקשת הכי זולה שיצאת מקודקוד u . הוא יבחר אותה לסייע הנוכחי של בורבוקה ולא יפתח את שאר החבילות. הוא עבורי לחיבור הבאה - רק כאשר יסיעים עם הקשתות בחיבור הנוכחה. הנכונות מגיעה מכך שהחזרה הקשתות בחבילות הבאות - גודלות מהקשנות בחיבורו שלו.
קודקוד יגדר עבורי חיבור שהוא כבר לא רלוונטי - אם ורק אם כל הקשתות שנמצאות בה גולו כוללות עצימות (קשנות שמחוברות לקודקוד באותו רכיב).

ניתוח זמן הריצה:

הועלות מרכיבת מ-3 חלקים:

1. **יעיוב מקדים:** לכל קודקוד $V \in u$ חישוב *Buckets* שלו: $O(\sum_{v \in V} \deg(v) \times \log(k)) = O(|E| \log k)$
2. **סריקה של קשתות זבל:** נבחן כי קשת זבל (שהתגלה כקשחת עצמית) נסרקת כאבל בדיק פעם אחדת - כשמগלים שהיא זבל מיסרים אותה. ולכן $O(1)$ פר קשת כלומר $O(|E|)$
3. **חיפוי קשת בכל סרב:** בכל אחד מהסבירים של בורבוקה כל קודקוד סורק חיבור אחד (או חלק ממנו). נבחן כי הקודקוד יודיע באיזו קבוצה עליו לחפש כי הוא העף כבר (חישבנו את החיבור סה"כ עבורי האלגוריתם) את הקשתות שהפכו לאבל. גודל כל חיבור הינו $\lceil \frac{\deg(v)}{k} \rceil$ ולכן עבורה עבור כל הקודדים בסבב היא בעלות:

$$O\left(\sum_{v \in V} \left\lceil \frac{\deg(v)}{k} \right\rceil\right) \leq \frac{|E|}{k} + |V| = O\left(\frac{|E|}{k} + |V|\right)$$

סה"כ ישים $(|V| \log |V|)$ סבבים אצל בורבוקה ולכן נקבל זמן הריצה הינו:

$$\log(|V|) \times \left(\frac{|E|}{k} + |V| + |E| \log k\right)$$

נדיר: $k = \lceil \log(|V|) \rceil$ ונקבל זמן ריצה:

$$O(|V| \log |V| + |E| \log(\log(|V|)))$$

המטרה: להפטר מ- $|V| \log |V|$.

נחלק את האלגוריתם ל-2:

1. נריץ את האלגוריתם הרגיל של בורבוקה (זה שעלותו $O(|E|)$ למציאת הקשתות סה"כ בסביב) לפחות $\log(\log|V|)$ סבבים בלבד. עלות חלק זה: $O(|E| \log \log |V|)$
2. בבדיקה: כיון שבסכל סיבוב מס' הקודקודים קטן פי 2 (פחות) קיבל כי אם נסמן ב- V' את גודל קבוצת הקודקודים לאחר ההרצות של בורבוקה אז היא:

$$|V'| = \frac{|V|}{2^{\log(\log(|V|))}} = \frac{|V|}{\log(|V|)}$$

2. כעת, נריץ את האלגוריתם של Yao על הגרף שנותר עם V' . נבחן כי $|V'| \leq |V|$ והאלגוריתם של יואו יעלה לנו:

$$|V'| \log |V'| + |E| \log(\log(|V'|))$$

$$\leq_* \frac{|V|}{\log |V|} \times \log(|V|) + |E| \log(\log(|V|)) = O(|V|) + O(|E| \log(\log(|V|))) = O(|E| \log(\log(|V|)))$$

הסביר לך: הפעלו את $|V'| = \frac{|V|}{\log |V|}$ רק על החלק $|V'|$ הראשון והשתמשנו בכך שהבקרה בשאר המיקומים של $|V'|$, ובחלק האחרון השתמשנו בכך ש($O(|V|) = O(|E|)$ כי הגרף V' קשור).

13.4 האלגוריתם של Karger – Klein – Tarjan

cutת נראת אלגוריתם שרצ בתוכלת ($O(|E|)$) זמן למציאת MST .

- הגדרה:** נסמן (V, E) יער על V (חלק מהגרף $G = (V, E)$: מזובר ביר על כלשהו של הגרף).
הגדרה: הינו u, v באוטו רכיב קשרות של F . נסמן $P_F(u, v)$ את המסלול הפשטוט הייחיד בין u ו- v ב- F . (קיימים מסלול יחיד בעץ)
הגדרה: $\{e|e \in P_f(u, v)\} = max_F(u, v)$ משקל הקשת הכבודה ביותר על המסלול הפשטוט מהל. כבודה ביותר ביחס ליחס סדר!)

הערה חשובה להמשך ומוטיבציה (איינטואיציה, זה אני הוסיףתי לא היה בהרצאה): אלגוריתם KKT מנסה להיות עצמן - במקומות לעבד קשה ולמצא את MST האמייתי של E הקשתות הוא דוגם חלק קטן מאוד ווקשות. הוא ימצא את MST שלו (יעיר F) והוא ישמש בו בצדיל לסנן את שאר הקשתות בגרף המקורי. זה יהיה יער שנוצר מדגימה אקראית. למען האינטואיציה עוד: נניח כי יש לנו גרף ענק, נזרוק לכל קשת מטבע: אם יצא עץ - היא תכנס לקבוצה בשם E_s , אחרת לא תכנס. cutת

נמצא את ה- MST של הקבוצה הקטנה זו, התוצאה היא יער שלו נקרא F .

הגדה: קשת $e = (x, y)$ כבודה אם:

1. x, y באוטו רכיב ל- F (למה? אם הם לא מוחוברים ב- F , אין לנו מסלול להשווות אלו ואז לא ניתן לומר בהכרח שהיא מיותרת).
 2. $e > \max_F(x, y)$.
- אחרת: נאמר כי e היא F קלה.

למה: היא e כבודה $\iff e \in MST$. (זה חלק גדול מלב האלגוריתם. האלגוריתם אומר שלמרות שהשווינו אותה רק לעיר חלקי F העובדה שהיא מעגל שבו היא כבודה מספיק בצדיו לומר: את לא בא MST בהכרח כי הוא היחיד)

קובסה שחורה: בהינתן F, V, E קיים אלגוריתם שモצא את כל הקשתות F כבדות בזמן לינארי $O(|E|)$.

טענה (משפט דגימה): בהינתן $G = (V, E_S)$, ניצר גוף אקראי ($H = (V, E)$ כך שכל קשת $e \in E$ נבחרת בהסתברות $1 - p < 0$. יהיה MSF (E) עיר פורש מינימום של H , איזה $\mathbb{E}[X] \leq \frac{|V|-1}{p}$ (באשר X מ"מ שסופר את מס' הקשתות ב- E שחן F קלות). ככלומר, אם דגמנו יער כאשר קשת נכנסת אליו בהסתברות p , וنبנה ממנה MSF , מס' הקשתות F קלות (אלו שלא הצלחנו לזרוק) הוא לכל היותר $O(\frac{|V|}{p})$.

הוכחה: נניח לצורך ההוכחה (או לא באמצעות הדרך שבה מצאנו את F !) כי השתמשנו בקורסקל על H על מנת למצוא את F . (זה לא אומר שהאלגוריתם שהרצנו הוא בהכרח של קروسקל! אך כאן זה מותר להניח זאת שכן ישנו MSF ייחד ולכן כל האלגוריתמים יחוירו את אותו מבנה בהכרח).

כלומר, קروسקל מורץ על הגראף הדגום H .

נתבונן על הרגע שבו קروسקל הסיף קשת $(x, y) \in F$. בזמנן אז, נסמן ב- A את הרכיב שמכיל את x וב- B את הרכיב שמכיל את y . בעת אנחנו יכולים להניח כי הקשת e הקללה ביחס בחתך H שמחברת את A ו- B והוא הקשת הקללה ביחס בחתוך (A, B) . נתבונן על כל הקשתות בגרף המקורי! שחוות את החתך בין A ל- B . חלק מהן נדגמו H' וחלק לא.

הקשת שקורסקל בחרה, היא הקשת הראשונה e מתוך הרשימה הניל' שאכן נדבמה לה. כמו קשתות היינו צריכים לדוגם עד שנגע לפחות הראשונה הזו? מדובר ב- $Geo(P)$ - בודקים עד להצלחה. תוחלת מספר הקשתות שנבדקו עד הצלחה הראשונה היא $\frac{1}{p}$. כל הקשתות יהיו לפני e ברשימה (אלו שלא נדגמו) הן בהכרח F קלות. מדוע? כאשר יער P ישלם, לא יהיה בין לבין מסלול שככל קשתות יותר (הר) הן בהכרח לא נכנסו! היא הראשונה שכנסה!).

הבחנה: כל הקשתות שמשמעותם ביחס הסדר לאחר e הן בהכרח F כבדות.

הסביר: ברגע שקורסקל הכניס את e ל- F , הוא חיבור את A ו- B . כל קשת אחרת e' בחתוך (A, B) שגדולה מ- $e: e' > e$: סוגרת בהכרח מעגל עם הקשתות שכבר בתוך A ו- B (הר) קשת שחוצה את החתך (a, b) היא קשת כך ש- $a \in A, b \in B$ וכן אם נסמן את הקשת הקללה בחתוך $(x, y) = e$ איזי ב- A ישנו מסלול P_{ax} וב- B ישנו מסלול P_{yb} ולכן $P_{ax} \circ (x, y) \circ P_{yb} \circ (b, a)$ הוא מעגל: שבו הקשת e' היא בהכרח הקשת הכבודה ביותר - ולכן בהכרח לא נכנסו! היא הראשונה ולא נמצאת באפ- MSF . מדובר בקשת כבודה!). ניתוח זה נכון לכל $e' > e$.

בעת, כל קשת שנכנסה ל- F (ישנן לכל היותר $\frac{1}{p}$ קשתות כבודה ב��ולת $|V| - 1$ כallo) ייצגה בתחולת $\frac{1}{p}$ קשתות לכל היותר שהיו רלוונטיות להיות פוטנציאליות קלות בחתוך שלה. קשתות שלא היו בחתוך של אף קשת הן בהכרח כבדות כפי שאמרנו. לכן e מגדירה $\frac{1}{p}$ קשתות קלות לכל היותר. לכן: מס' הקשתות הקלות בתחולת כל היותר חסום ב:

13.5 פסאודו של KKJ

$:ALG(V, E)$

א. אם $E = \emptyset$ החזר \emptyset .

ב. הרץ 2 סבבים של $Boruvka$ (נשארו $\frac{|V|}{4}$ קודוקדים לכל היותר) לקבלת עיר קשותות B .

ג. כווץ קשותות B ב- G . נסמן את הגרף המכווץ ($G' = (V', E')$)

ד. כל קשת M' מצטרפת ל- E_S בסיכוי $p = \frac{1}{2}$ (p (דגימת קבוצת הקשותות נשארו לאחר הכיווצים)

ה. $F = ALG(V', E_S)$ (מדובר בעיר פורש מיניימום על הקשותות שדגמוני)

ו. מצא את כל הקשותות h ב- F (באמצעות קופסה שחורה) ומחק אותן מ- E' .

ז. נסמן L את הקשותות שנשארו ב- G' (נותרו בתוך $|V'| - 1$ $\approx \frac{|V'| - 1}{2}$)

הערה. למעשה בדוגמה הקודמת כל מי שנכנסות אל L הן אלו שמתוחת ל- e ואלו שלא הן בהכרח

ה כבודות שייעפו ולא יכנסו. נבחן כי בהכרח קשותות L מצלות MSF (וודע קשותות!)

ח. $F' = ALG(V', L)$

ט. החזר $F' \cup B$

האלגוריתם מוחזר לבסוף MST כיון שהגרף המקורי במקור הוא קשור מהנהנתנו והעיר הפורש המיניימלי שלו הוא עצם.

הסבר:

א. בבירור: אם אין קשותות, אין מה לחבר בגרף. תנאי עצירה.

ב+ג: עושים מעין "דיאטה" לקודוקדים. אנחנו מרים שניי הרצות של בורובקה. מדוע דוקא 2?

זה לא קרייטי דוקא 2 - יוכלו גם 3 וכיו אך חשוב לא הרצאה אחת. זה יהיה חשוב לניטוח זמן הרצאה בתוחלת בהמשך. נסמן B את הקשותות שboroboka בוחר: לפי תוכנות החתק - הן בהכרח MST (ונכונותו של בורובקה, אלו בהכרח קשותות שנמצאות ב- MST). נבחן כי לאחר שתי הרצות נקבל שבגרף שקבוצת קודוקדי (המכיווצים) היא לכל היותר בגודל $\frac{|V|}{4}$. כתעת: עוברים לעובוד על גראף הרבה יותר קטן מבחינת קודוקדים.

ד: כתעת מבצעים דוגמה: S . כל קשת M' (אלו שנותרו בגרף המכווץ) ת策טראף לגרף הדוגימה E_S בסתירותו $\frac{1}{2}$.

ה: מבצעים רקורסיה: מבקשים מהאלגוריתם למצוא את MSF של הקבוצה הקטנה זו. שוב ושוב - עד שנגיע לכך שאין קשותות.

בסיום שלב ה, בידינו MSF (עיר פורש מיניימלי) של הקשותות שדגמוני. F הוא לא MST של הגרף המקורי! הוא MST של חלק מהקשותות (בתוחלת, חצי מהקשותות) אותן חרכנו באקרה.

ו: כתעת, יש בידינו MSF שהוא, אנחנו יודעים למי מה שטענו קודם כי הקשותות הכבודות לא יהו בו. באמצעות הקופסה שחורה - אנחנו מוצא את כל הקשותות ופושט נעל'ן אותן מ- E' . למעשה: אנחנו יודעים לזאת על הרבה קשותות כרגע שאין ב- MST הסופי.

בשלב זה - הקשותות M' שללא נבחרו (בתוחלת חצי מהן לא נבחרו) חוזרות לתמונה. קודם לנו הרצינו עיר בלבדין, כאן הן חוזרות לתמונה.

ז: כל הקשותות שנשארו לאחר המהיקה, יסומנו ב- L . אלו שלא הצליחנו לזרוק - ישן בתוחלת $(|V'|)$ $O(|V'|)$ כאמור, מדובר דليل מאד של קשותות! לינארי במספר הקודוקדים.

ח: בסוף, כישיש לנו גראף רזה מאד, מרים עליו את האלגוריתם עם קבוצת הקשותות שלא הצליחנו למחוק L ו- V' .

ט: התוצאה מהרקורסיה תהיה MSF האמתי של הגרף המכווץ.

סה"כ: B אלו הקשותות שאנו יודעים שהכרח MST מנכונות בבורסקה, ו- F' זו תוצאה על הגרף המכווץ (לאחר הרצת פעמיים של בורסקה) ולכן סה"כ העץ שלנו מורכב מ- $B \cup F'$.

13.6 זמן הריצה

על מנת לנתח את זמן הריצה נסמן $|V| = n, |E| = m$

$$T(n, m) = O(m + n) + T(n_1, m_1) + T(n_2, m_2)$$

נבחין כי כל הפעולות הן בזמן לינארי פרט לרוקורסיות כל העבודה שמתבצעת לינארית ותסומן ע"י $O(n, m)$. $T(n_1, m_1)$ הוא הקריאה הראשונה לרוקורסיה ו- $T(n_2, m_2)$ הוא הקריאה השנייה לרוקורסיה. נרצה לחשב את $\mathbb{E}[T(n, m)]$

$$\mathbb{E}[T(n, m)] = O(m + n) + \mathbb{E}[T(n_1, m_1)] + \mathbb{E}[T(n_2, m_2)]$$

ונכיה באינדוקציה כי $\mathbb{E}[X] \leq 2 \times a(\bar{m} + 2\bar{n})$ כאשר $a > 1$ קבוע ($O(m + n) \leq a(m + n)$ כלומר).

משפט 6. מתקיים $\mathbb{E}[T(n, m)] \leq 2a(\bar{m} + 2\bar{n})$.

הוכחה: נוכיח את הטענה באינדוקציה על $m + n$.

בסיס: טריוויאלי.

צעד: נכוחות לכל $n + m < n' + m'$. נוכיח עבור $n + m < n' + m'$.

$$\mathbb{E}[T(n, m)] = \mathbb{E}[a(m + n)] + \mathbb{E}[T(n_1, m_1)] + \mathbb{E}[T(n_2, m_2)]$$

$$\leq_* a\bar{m} + a\bar{n} + 2a(\bar{m}_1 + 2\bar{n}_1) + 2a(\bar{m}_2 + 2\bar{n}_2) =$$

* הוא הנחת האינדוקציה.
נניח כי $\frac{\bar{n}}{4} \leq \bar{n}_1, \bar{n}_2 \leq \frac{\bar{n}}{2}$ (אպעויות שני סכמי בובוקה)
וכן $\frac{|E'|}{2} \leq \frac{|E|}{2} \leq \frac{\bar{m}}{2}$ (קשותות כניסה בהסתגרות חצי).
וכן $\frac{\bar{n}}{2} \leq \bar{n}_2 \leq 2 \times \frac{\bar{n}}{4} = \frac{\bar{n}}{2}$ (לי משפט הזוגית).
cutet נציג:

$$\leq a\bar{m} + a\bar{n} + 2a\left(\frac{\bar{m}}{2} + \frac{\bar{n}}{2}\right) + 2a\left(\frac{\bar{n}}{2} + \frac{\bar{n}}{2}\right) =$$

$$= 2a(\bar{m} + 2\bar{n})$$

כנדרש.

מסקנה: זמן הריצה של האלגוריתם $O(n + m) = O(m) = O(|E|)$ בתחילת.

בהצלחה לככלנו ב מבחנים!

14 סיכום אלגוריתמים שראינו בקורס + זמן ריצה ("קופסאות שחורות")

אלגוריתם למכפלת פולינומיים:

אלגוריתם FFT: בהינתן שני פולינומיים, A ו- B מדרגה חסומה n , מחשב את המכפלה $O(n \log n)$.

אלגוריתם מרחק האמינה: בהינתן מחרוזת P באורך n ובתבנית בגודל m מוחזיר את מרחק ההאמינה של המחרוזות מהתבנית עבור כל היסט שלה. זמן הריצה $O(n \log m)$.

אלגוריתם למציאת MST:

1. האלגוריתם של פרים - משתמש בתור קדימות ועולתו באמצעות עץ פיבונacci הינה $O(|V| \log |V| + |E|)$

2. האלגוריתם של ק魯סקל - משתמש ביןין פיניד ועולתו $(O(|E|(\alpha|V|) + \text{sort}(|E|)))$

אלגוריתמי גרפים ומסלולים קצרים ביותר:

אלגוריתם BFS: מטפל בSSSP במקורה הלא ממושקל. סיבוכיות זמן הריצה שלו $O(|E| + |V|)$

אלגוריתם DFS: סריקה לעומק של הגרף G , בזמן $(|V| + |E|) O(|E|)$. אפשר למשל בדיקה של האם קיימים בגרף מעגל (לאחר הריצת האלגוריתם נבדוק האם יש קשתות אחוריות, אם כן אז יש מעגל).

אלגוריתם למציאת מעגל בגרף: מוציאים DFS ובודקים במהלך הרצה האם יש קשתות אחוריות. אם מצאנו צ'ז: דוחה על מעגל אחרתי, אין. זמן ריצה $(|E| + |V|)$

אלגוריתם למציאת רכיבי הקשרות חזקה: בגרף מכון, רוצים למצוא את רכיבי הקשרות חזקה. בזמן $O(|E| + |V|)$

אלגוריתם מיון טופולוגי: לכל קשת (v_i, v_j) יתקיים בסוף המיוו $j < i$. זמן הריצה $O(|E| + |V|)$

אלגוריתם גרען דו צדדי: בהינתן גרען $G = (V, E)$ דו"צ, מוחזיר את הקבוצות L, R .

(מבחן את הטענה $i \in L \iff \forall j \in R : (i, j) \in E$)

אלגוריתם למציאת זמן ריצה SSSP: בהינתן גרען חסר מעגלים, מחשב SSSP באמצעות תכנות דינמי בסיבוכיות זמן ריצה $O(|E| + |V|)$

האלגוריתם של בלמן פורד: מטפל בSSSP במקורה הממושקל. מניח שיתכנו משקלים שליליים

אך לא יתכוו מעגלים שליליים. סיבוכיות זמן הריצה שלו $O(|E| \times |V|)$

האלגוריתם של דיקסטרה: מטפל בSSSP במקורה הממושקל, אך מניה שלא יתכוו משקלים שליליים. סיבוכיות זמן הריצה שלו $O(|V| \log |V| + |E|)$ עם פיבונacci ועם עירומה ביןארית $O(|V| \log |V| + |E| \log |V|)$

האלגוריתם של פלייד וורש: מטפל בAPSP במקורה הממושקל, מניח שיתכנו משקלים שליליים אך לא יתכוו מעגלים שליליים. עובד בתכונות דינמי עם נוסחת נסיגה פשוטה למדי, ומוחזיר את $D^{(n)}$ מטריצת המרחוקים ו- π מטריצת הקודומים. רץ בסיבוכיות זמן ריצה $O(|V|^3)$

האלגוריתם של ג'ונסון: מטפל בAPSP במקורה הממושקל, מניח שיתכנו משקלים שליליים אך לא יתכוו מעגלים שליליים. "מרמה" ומשתמש באלגוריתמים קיימים. תחילת מרץ בלונו פרוד. אס מצא מעגל שלילי מפסיק מיד. לאחר מכן מנסה (u, v, δ) שחייב בלמוני פרוד למען הגדרת פונקציית משקל על הקודדים, ובאמצעותה פונקציית משקל חדשה על הקשותות. פונקציית המשקל המוגדרת החדש \hat{w} היא אי שלילית ולכן הוא מרי $|V|$ פעמים דיקסטרה. סה"כ זמן הריצה $O(|V||E| + |V|^2 \log |V|)$

אלגוריתם לחישוב הסגור הטרנסיזבי של גרען G^* : משתמשים בAPSP (של פלייד וורש למשל) ומקבלים זמן ריצה של $O(|V|^3)$

אלגוריתם לחישוב סגור טרנסיזבי של גרען מכון G^* באמצעות כפל מטריצות: משתמשים בהפרד ומשול ומקבלים זמן ריצה של $O(|V|^\omega)$

אלגוריתם כפל מהיר (FMM): מחשב את תוצאה המכפלה של שתי מטריצות A, B בעלות זמן $2 \leq \omega \leq 2.37287$ $O(n^\omega)$ באשר ω מטריצות A, B

האלגוריתם של סידל: מוחשב את $APSP$ במקורה הלא ממושקל באשר הגרף G לא מכוען, מכיון
מטריצה $\Delta_{i,j} = \delta(v_i, v_j)$ בסיבוכיות ז閏ון $O(|V|^{\omega})$.

רשותות זרימה:

השיטה של פורץ-פלקריםון: מוצאת את הזרימה המקסימלית ברשת זרימה $(V, E) = G$ נתונה.
שכל למציאות חתך (s, t) מינימלי. מניח שכל הקיבולות הם ערבים שלמים. עלות ז閏ון הריצה היא
 $O(|f^*| \times |E|)$ באשר $|f^*|$ הוא גודל הזרימה המקסימלי.

האלגוריתם של אדמונס קארוף: מוצאת זרימה מקסימלית בסיבוכיות $O(|V|^2 \times |E|^2)$
אלגוריתם למציאת זיוג מקסימום בגרף דו צדדי: בונה גרף G' וטוען שזרימת המקסימום בו
מתאימה זיוג מקסימום, וכן על ידי שימוש בפורץ פלקסון כיוון ש $= |V| \leq |L| = |M^*| = |f^*|$.
 $O(|V|)$ ז閏ון ריצת האלגוריתם ($O(|E||V|)$).

אלגוריתם למיניאת חתך (S, T) מינימלי: אכן לפ' $MPMC$ מתקיים שערך הזרימה המקסימלית
הוא ערך החתך המינימלי אך רוצים למציאה את החתך. האלגוריתם מרים אלגו למציאת M ולאחר
מכן מגדיר כמה הדרגות, סה"כ ז閏ון ריצתו כזמנם ריצה למציאת זרימה מקסימלית. למשל $O(|V|^2 \times |E|)$.

האלגוריתם של Dinic: מוצאת זרימה מקסימלית בסיבוכיות $O(|V|^2 \times |E|)$
האלגוריתם של Hopcroft – Karp: מקבל רשת זרימה עם קיבולות על הקודקודים, בונה גרף
חדש G' עם קיבולות על הקשתות בלבד, ומירץ ב $\sqrt{|V|}$ פעימות ראשונות את דיניצ' ובשאר את פורץ
פלקסון. מקבל זרימה מקסימלית בגרף זה בז閏ון $(\sqrt{|V||E|})O$. חשוב להזכיר - הקיבולות על
הקודקודים הם בדיקות 1.

כל כביד:

אלגוריתם לייהוי משולשים - מחלק את המשולשים לשני סוגים - קלים וכבדים ורץ בסיבוכיות ז閏ון
 $O(|E|^{1.5})$.

אלגוריתם מרחק האמיג לא"ב כלילי - משתמש בשני אלגוריתמים שונים ומוסוג אותן בהתאם באמצעות
מס' המופעים. אותן כבידה היא אותן שמויפה יותר מ $\sqrt{mlogm} = c$ פעמים. ז閏ון ריצת האלגוריתם
 $O(n\sqrt{mlogm})$.

מציאת משולשים בגרף דיל (ובכל גраф, עדיף בדיל): מחלקים לכבדים וקלים, באשר על הכלבים
מריצים FMM ועל הקלים את הנאיבי (על כל זוגות השכנים) ומגעים לזמן ריצה $(\tilde{O}(|E|^{\frac{2w}{w+1}}))$.

אלגוריתמים רנדומיים:

אלגוריתמי מונטה קרלו:

א. אלגוריתם ליזוא בפטל מטריצות:

1. אלגוריתם בסיסי - מסתמך על העובדה שאם $Cv = ABv$ אז $C = AB$ ובודק את תוכאות
הוקטור. סיבוכיות ז閏ון ריצה $O(n^2)$ וסיכוי לטעות $\geq \frac{1}{2}$.

2. האלגוריתם השני - מרים את האלגוריתם הראשון ($n = \alpha log(n)$) k פעמים, סיבוכיות ז閏ון ריצה
הינה $O(n^2 log(n))$ והסיכוי לטעות הוא לכל היתר $\frac{1}{n^\alpha}$ באשר $2 \geq \alpha$ קבוע.

אלגוריתמי לאס וגאס:

א. אלגוריתם מין מהיר - הוכחנו בקורס כי מס' ההשואות של אלגוריתם מין מהיר קלסטי
בתוחלת הינו $O(nlogn)$.

ב. אלגוריתם מין מהיר פרנוואידי - אלגוריתם מין מהיר שמודא בכל שלב כל צד בחר לפחות
 $\frac{n}{4}$ יברים. הוכחנו כי ז閏ון ריצה שלו בתוחלת הינו $O(nlogn)$.

מין דלי: אלגוריתם דטרמיניסטי שמקבל קלט רנדומי, מחלק את הקלט לבאקטים שונים ובההתאם לכך
מבצע בכל באקט מין הכנסה ולבסוף משריר את הבאקטים. תוחלת ז閏ון ריצה של המין הינו $O(n)$.

בעיית הצביעה: