TSP

2η προσέγ.

$$\frac{\sum_{e \in T} p_e}{\sum_{e \in T} w_e} \geq c$$

(1) KNAPSACK

(2) Binary search αν υπάρχει δέντρο κάθε ακμή έχει νέο κόστος $c'_e = p_e - c\, w_e$ → θετικό γίνεται

Ποιο το μέγιστο c ;

Δέντρο που κάνει max το $\sum_{e \in T} p_e = p^*$ ακέραιος (ώστε να έχω ακέραιους)

— // — min το $\sum_{e \in T} w_e = w^*$

$C_{max} = \frac{p^*}{w^*}$ $\quad C_{min} = 0$ ή κάτι αντίστοιχο

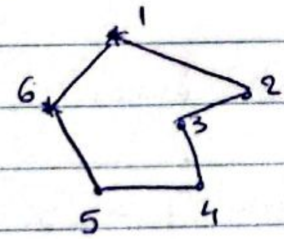
- Θέλω λόγο που να αντιστοιχεί σε δέντρο

Binary search → θέλω δέντρο που πετυχαίνεται (κρατάω ακμές) → γιατί μπορεί να μην πετυχαίνεται το δέντρο που βρίσκει η binary search (πολύ κοντά αλλά όχι ακριβώς)

* Προσοχή σε υπερχείλιση: διαιρέσεις double

$d: N \times N \to \mathbb{R}_+$

αποστάσεις ίσως $d(i,j) \neq d(j,i)$ (όχι συμμετρία)



Ποιά η τελευταία πόλη j πριν την 1η (έχω κύκλο),

~~Απόστ~~ Μήκος περιοδείας 1 → καλύπτει τις υπόλοιπες → j → 1

Έχω δρόμους από κάθε πόλη σε κάθε πόλη → πλήρες γράφημα

Δε 20/11/2023

* Length ($\underbrace{1 \to N \setminus \{1, j\} \to j}_{\text{περιοδεία}}$) + d(j, 1) = βέλτιστο μήκος περιοδείας

↓ ελάχιστο

Bottom up

Ελάχιστο μήκος περιοδείας $L(S, j)$ — ποιές κορυφές έχει καλύψει — τελευταία κορυφή

μήκους = "μερικής"

$1 \to S \to j$ — δεν έχει 1, j

Διαφ 33 λίγο διαφορ.

$(*) = \min_{i \in S} \{ L(S \setminus \{i\}, i) + d(i, j) \} = L(i, S)$ — n, 2^n

$n \cdot 2^n$ χρόνος εκτέλεσης αναδρομής → $n^2 \cdot 2^n$

exhaustive search → όλες πιθανές περιπτώσεις

Εδώ για κάθε περίπτωση (→ υποσύνολο (όχι μετάθεση)) παίρνω βέλτιστο μήκος (→ υπάρχουν πολλές, αλλά κρατάω ένα)

{ δεξιό της αναδρομής

20 σημεία $20! = 2.4 \times 10^{18}$

$20^2 \cdot 2^{20} = 4.2 \cdot 10^8$!!!!

EDIT DISTANCE

Διαφορετ. χαρακτήρες → κόστος

Παράλειψη → μικρότερο/άλλο κόστος

Μοιάζει με longest common subsequence

Σαρώθηκε με το CamScanner

$$LCS(x_1,\dots,x_n,y_1,\dots,y_m) = \begin{cases} LCS\begin{pmatrix} x_1 \dots x_{n-1} \\ y_1,\dots y_{m-1} \end{pmatrix} + \underbrace{1}_{\text{ο ίδιος χαρακτ.}} & x_n = y_m \\ \max\begin{Bmatrix} LCS\begin{pmatrix} x_1,\dots,x_{n-1} \\ y_1,\dots,y_m \end{pmatrix} \\ LCS\begin{pmatrix} x_1 \dots x_n \\ y_1 \dots y_{m-1} \end{pmatrix} \end{Bmatrix} & \text{αλλιώς} \end{cases}$$

$\downarrow$

$\Theta(n \cdot m)$

Τώρα θέλω min κόστος

$$Cost(x_1,\dots,x_n,y_1,\dots,y_m) = \min \begin{cases} Cost\begin{pmatrix} x_1 \dots x_{n-1} \\ y_1 \dots y_{m-1} \end{pmatrix} + d(x_n, y_m) \quad \text{όχι αναγκαία ίσοι} \\ cost\begin{pmatrix} x_1,\dots,x_{n-1} \\ y_1,\dots,y_m \end{pmatrix} + \delta \quad \text{βάζω κενό} \\ cost\begin{pmatrix} x_1,\dots,x_n \\ y_1,\dots,y_{m-1} \end{pmatrix} + \underbrace{\delta}_{\text{κόστος κενού}} \end{cases}$$

Ενδιαφέρον $\rightarrow$ $O(n+m)$ space

String Matching CLRS

συμβολοσειρά: πίνακας με χαρακτήρες

$T = [a_1, \dots, a_n]$ text

$P = [P_1, \dots, P_m]$ pattern $\boxed{\text{Συνήθως } m \ll n}$

1η συμβολοσειρά / εμφάνιση του pattern στο text

Ts: μετατόπιση

$a_1, a_2, \dots, \underline{a_s}$ από εδώ ks πέρα βρίσκω το pattern

1η λύση DFA που αναγνωρίζει το pattern (αν έχω συγκεκριμένη συμβολοσειρά)

Προσοχή αν έχω CCAN και απορρίπτω το CCA δεν θα βρω το CAN

Strings as Numbers
μετατροπή σε δεκαδικό αριθμό (Διαφ 6)

Κανόνας Horner

Αναδρομικό από το t_s θέλω το t_{s+1}
Διαφ 7

t_s
2 1 8 3 6 7 9
t_{s+1}

	0	1	2	3	4	5	6	7
s	a	b	c	d	a	b	c	a
p	0	0	0	0	1	2		

p[0] = 0   i = 1 , j = 0
για pattern

s[1] ≠ s[0]   p[1] = 0
i = 2   j = 0
s[2] ≠ s[0]   p[2] = 0
i = 3   j = 0
s[3] ≠ s[0]   p[3] = 0
i = 4   j = 0
s[4] = s[0]   p[4] = j+1 = 1
i = 5   j = 1
s[5] = s[1]   p[5] = j+1 = 2 ← μεγαλύτερο suffix που είναι και prefix
i = 6   j = 2
s[6] = s[2]   p[6] = 3 = j+1
s[7] ≠ s[3]   j = p[j-1] = 0
s[7] = s[0]   p[7] = j+1 = 1