

Resumen de teoremas para el final de Matemática Discreta II

Agustin Curto, agucurto95@gmail.com

2016

Índice general

1. Parte A	2
1.1. La complejidad de EDMONS-KARP	2
1.2. Las distancias de Edmonds-Karp no disminuyen en pasos sucesivos	2
1.3. La complejidad de DINIC	2
1.4. La complejidad de WAVE	4
1.5. La distancia entre NA sucesivos aumenta	4
2. Parte B	5
2.1. 2-COLOR es polinomial	6
2.2. Probar que:	6
2.2.1. El valor de todo flujo es menor o igual que la capacidad de todo corte.	6
2.2.2. Si el valor de un flujo es igual a la capacidad de un corte entonces el flujo es maximal y el corte minimal.	6
2.2.3. Si un flujo es maximal entonces existe un corte con capacidad igual al valor del flujo.	6
2.3. Complejidad del Hungaro es $\mathcal{O}(n^4)$	6
2.4. Teorema de Hall	6
2.5. Teorema del matrimonio	6
2.6. Si G es bipartito $\Rightarrow \chi'(G) = \Delta$	6
2.7. Teorema cota de Hamming	6
2.8. Sea H una matriz de chequeo de un código C , pruebe que:	6
2.8.1. $\delta(C)$ = mínimo número de columnas linealmente dependientes de H	6
2.8.2. Si H no tiene la columna cero ni columnas repetidas $\Rightarrow C$ corrige al menos un error	6
2.9. Sea C un código cíclico de dimensión k y longitud n y sea $g(x)$ su polinomio generador, probar que:	6
2.9.1. C está formado por los múltiplos de $g(x)$ de grado menor a n	6
2.9.2. El grado de $g(x)$ es $n - k$	6
2.9.3. $g(x)$ divide a $1 + x^n$	6
3. Parte C	7
3.1. $4\text{-COLOR} \leq_p \text{SAT}$	7
3.2. 3-SAT es NP-Completo	7
3.3. 3-COLOR es NP-Completo	7

Capítulo 1

Parte A

1.1. La complejidad de EDMONS-KARP

Teorema: La complejidad de $\langle E - K \rangle$ con $n = |V|$ y $m = |E|$ es $\mathcal{O}(nm^2)$.

Prueba: Probaremos primer un hecho por inducción para enunciarlo, sean:

f_0, f_1, f_2, \dots la sucesión de flujos creados por $\langle E - K \rangle$.

El paso k es el paso que crea f_k . Definimos funciones:

- $d_k(x) = \text{distancia}^1$ entre s y x en el paso k en caso de existir, si no ∞ .
- $b_k(x) = \text{distancia}$ entre x y t en el paso k en caso de existir, si no ∞ .

Observación:

- $d_k(s) = 0$
- $b_k(t) = 0$

Lema interno: queremos probar que

1. $d_k(x) \leq d_{k+1}(x)$
2. $b_k(x) \leq b_{k+1}(x)$

Prueba: lo probaremos por inducción y solo para d_k ya que para b_k la prueba es análoga.

$$\text{HI: } H(i) = \{\forall_z : d_{k+1}(z) \leq i, d_k(z) \leq d_{k+1}(z)\}$$

1. $H(0) = \{\forall_z : d_{k+1}(z) \leq 0, d_k(z) \leq d_{k+1}(z)\}$

Pero $d_{k+1} \leq 0 \Rightarrow$

1.2. Las distancias de Edmonds-Karp no disminuyen en pasos sucesivos

1.3. La complejidad de DINIC

Teorema: La complejidad del algoritmo de Dinic es $\mathcal{O}(n^2m)$.

Prueba: Como la distancia entre s y t en networks auxiliares consecutivos aumenta y puede ir a lo sumo entre 1 y $n - 1$ entonces hay a lo sumo $\mathcal{O}(n)$ networks auxiliares.

Notación: llamemos PB al proceso de hallar paso bloqueante en un network auxiliar con Dinic.

Luego la complejidad de Dinic es $\mathcal{O}(n) \text{ compl}(\text{PB})$. Para probar que la complejidad de Dinic es $\mathcal{O}(n^2m)$ debemos probar que $\text{compl}(\text{PB}) = \mathcal{O}(nm)$.

Para hallar un flujo bloqueante:

1. Crear un NA: Como es con BFS es $\mathcal{O}(m)$
2. Hallar bloqueante, denotemos:
 - A: avanzar
 - R: retorceder
 - I: inicializar e incrementar

El paso bloqueante de Dinic luce de la forma:

AA...AIAAARA...AIAARAAARR...IA...

subdiviámoslo en palabras del tipo:

- AA...AI
- AA...AR

donde las primeras son todas A pudiendo ser 0 la cantidad de la misma. Debemos determinar:

1. Cual es la complejidad de cada subpalabra.

Recordemos que:

- A: $\{ P[i + 1] = \text{algún elemento de } \Gamma^+(P[i]) \}$
 $i = i + 1$
 \Rightarrow A es $\mathcal{O}(1)$
- R: $\{ \text{borrar } P[i - 1] \text{ del NA} \}$
 $i = i - 1$
 \Rightarrow R es $\mathcal{O}(1)$
- I: $\{ \text{Recorre dos veces un camino de longitud } d = d(t) \}$
 \Rightarrow R es $\mathcal{O}(d)$

Por lo tanto:

$$\text{compl}(A...AR) = \mathcal{O}(1) + ... \mathcal{O}(1) + \mathcal{O}(1) = \mathcal{O}(j) \quad (1.1)$$

Pero como cada A hace $i = i + 1$ y tenemos $0 \leq i \leq d \Rightarrow j \leq d$.

$$\therefore \text{compl}(A...AR) = \mathcal{O}(d)$$

Similarmente:

$$\text{compl}(A...AI) = \mathcal{O}(1) + ... \mathcal{O}(1) + \mathcal{O}(1) = \mathcal{O}(d) + \mathcal{O}(d) = \mathcal{O}(d) \quad (1.2)$$

Pero como cada A hace $i = i + 1$ y tenemos $0 \leq i \leq d \Rightarrow j \leq d$.

$$\therefore \text{compl}(A...AR) = \mathcal{O}(d)$$

2. Cuantas palabras hay de cada tipo.

1.4. La complejidad de WAVE

1.5. La distancia entre NA sucesivos aumenta

Capítulo 2

Parte B

2.1. 2-COLOR es polinomial

2.2. Probar que:

2.2.1. El valor de todo flujo es menor o igual que la capacidad de todo corte.

2.2.2. Si el valor de un flujo es igual a la capacidad de un corte entonces el flujo es maximal y el corte minimal.

2.2.3. Si un flujo es maximal entonces existe un corte con capacidad igual al valor del flujo.

2.3. Complejidad del Hungaro es $\mathcal{O}(n^4)$

2.4. Teorema de Hall

2.5. Teorema del matrimonio

2.6. Si G es bipartito $\Rightarrow \chi'(G) = \Delta$

2.7. Teorema cota de Hamming

2.8. Sea H una matriz de chequeo de un código C , pruebe que:

2.8.1. $\delta(C)$ = mínimo número de columnas linealmente dependientes de H

2.8.2. Si H no tiene la columna cero ni columnas repetidas $\Rightarrow C$ corrige al menos un error

2.9. Sea C un código cíclico de dimensión k y longitud n y sea $g(x)$ su polinomio generador, probar que:

2.9.1. C está formado por los múltiplos de $g(x)$ de grado menor a n

2.9.2. El grado de $g(x)$ es $n - k$

Capítulo 3

Parte C

3.1. 4-COLOR \leq_p SAT

3.2. 3-SAT es NP-Completo

3.3. 3-COLOR es NP-Completo

Bibliografía

- [1] CURTO AGUSTÍN , «Matemática Discreta II, apuntes de clase», *FaMAF, UNC*.
- [2] MAXIMILIANO ILLBELE, «Resumen de Discreta II, 16 de agosto de 2012», *FaMAF, UNC*.