CS499 Homework 11 (First Draft)

Intersteller

Exercise 11.1

As we have proved in **Exercise 8.8**, the largest antichain of $\{0, 1\}^n$ is $\binom{n}{\lfloor n/2 \rfloor}$.

We define a layer as a set of strings containing same number of $'1'$ and is sorted by how many $'1'$ a string in this layer contains.

1.There are $\binom{n}{\lfloor n/2 \rfloor}$ strings in the middle layer, which has the most strings. Since any two strings from the same layer are not comparable, there are at least $\binom{n}{\lfloor n/2 \rfloor}$ chain partitions.

2. All strings in any layer except the middle one can form chains with unique strings in its adjacent layer with the following method:

Since $0 \leq k < n/2$, there are more $'0'$ than $'1'$ in this layer, we calculate a strings score by the following rules: scan the string from the beginning and the initial score is 0, add one if current digit is 0,minus one otherwise. Find the digit where the first highest score appears (which must be a $'0'$),change it to 1. Then we get a string belongs to its adjacent layer and these two strings can form a chain(they are comparable).Now we prove that this string is unique:

Assume that there are two different strings that transform into a same string. Assume that the first string changes the i-th digit, and the other changes the j-th digit (with no loss of generality, assume $i < j$). Then the i-th digit of the second string and the j-th digit of the first string are 0, whereas other digits are the same. Assume that the score of the $(i-1)$-th digit is k.Then the score of the i-th digit is $k+1$ for the first string and $(k-1)$ for the second. Assume that the score of the $(j-1)$-th digit for the first string is $k+1+p$,then the score of the $(j-1)$-th digit for the second string is $k-1+p$.The score of the j-th digit for the second string is $k+p$. Since the changing digit is where the first largest score occurs, we have

$$k + 1 \geq k + 1 + p$$

$$k < k + p$$

where we get $p \leq 0$ and $p > 0$ which contradict each other. So different strings cannot transform into a same string by the method.

Accordingly, we can get a matching of size $\binom{n}{k}$ between the k-th layer and the $(k+1)$-th layer.
Exercise 11.2

This case, $|\Gamma(A)| \geq |A|$ for every $A \subseteq L$. We use contradiction to prove it. Assume $|\Gamma(A)| < |A|$, $\exists A \subseteq L$. Since every vertex has degree d, consider A, we have $d \times |A|$ edges. Since $\frac{d \times |A|}{|\Gamma(A)|} > d$, there must exist a vertex in R which degree is larger than d. It is a contradiction. Therefore, $|\Gamma(A)| \geq |A|$ for every $A \subseteq L$. According to course video, if $|\Gamma(A)| \geq |A|$ for every $A \subseteq L$, then (and only then) there exists a matching of size $|L|$. G has a perfect matching.

Exercise 11.3

Based on the mathematical induction, we have:

(1) When $d = 1$, obviously the edges $E(G)$ can be partitioned into d perfect matchings.

(2) When $d > 1$, we suppose when $d = n$, the edges $E(G)$ can be partitioned into d perfect matchings. When $d = n + 1$, we can find a matching M of size $|L|$ based on the 11.2, now we delete M from $E(G)$ and we get a new bipartite graph $G\prime$ whose every vertex has degree n. Based on the inductive assumption, the edge $E(G\prime)$ can be partitioned into n perfect matchings. We can add M to $M_1, \cdots, M_n$ and get matchings $M_1, \cdots, M_{n+1} \subset E(G)$ such that $(1) M_i \cap M_j = \emptyset for 1 \leq i < j \leq n+1$ and $(2) M_1 \cup M_2 \cup \cdots \cup M_{n+1} = E(G)$.

Therefore, if G is a d-regular bipartite graph, the edges $E(G)$ can be partitioned into d perfect matchings.

Exercise 11.4

1. For every vertex $X \in V$, add $X\prime$ to constitute $V\prime$. If the capacity of X is $c(X)$ in G, the capacity of $edge(X, X\prime)$ in $(G\prime)$ is $c(X)$. If $edge(X, Y) \in E$, $edge(X\prime, Y) \in E\prime$ and $c(X\prime, Y)$ is ∞.

2.

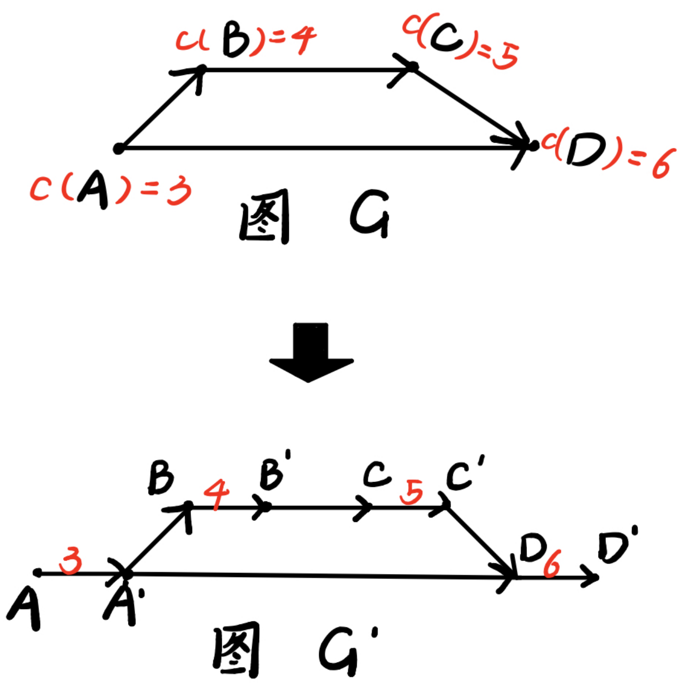


Figure 1:

3. We can set the capacity of every vertex in G as 1 and transform $G = (V, E, c)$ into a network $G\prime = (V\prime, E\prime, c\prime)$ by the method we give in 11.4.1. Then we use Edmonds-Karp algorithm on the network. If the maximum flow is k, we find k paths $p_1, p_2, \cdots p_k$, each from s to t, such that the paths are internally vertex disjoint.

Exercise 11.5

In the i-th layer, there are $\binom{n}{i}$ vertexes, for each vertex, we can form a path that ends at the $(n-i)$-th layer with the method we show in **Exercise 11.1**. When we reach a layer with more '1' than '0', we can still use this method to form a matching for the following reason. Since we begin at the i-th layer($i < n/2$), we have $(n - i)$ '0's and i '1's, so the score of the last digit equals $(n - 2i)$, each time we reach the next layer, the max score minus 1 for the digits before the changing digit remains unchanged. Totally we cross $(n - 2I - 1)$ layers from the i-th layer to $(n - i)$-th layer, so before we reach the $(n - i)$-th layer, the max score of every string in such path is always positive, which means the algorithm is feasible. Therefore, we can find $\binom{n}{i}$ paths as requested.

Exercise 11.6

1. e is always full. b, c, d, f, g, h, i is sometimes full. a is never full.

2. e is always crossing. a, b, c, d, f, g, h, i is never crossing.

Exercise 11.7

It is impossible to be always full and never crossing.

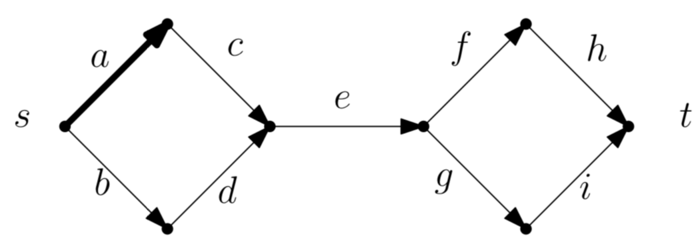
Proof: Consider an edge e which is always full. We assume the max flow is MAX and $f(e) = c(e)$. We delete this edge and the max flow of residual network is $MAX - c(e)$. There must exist minimum cut $MAX - c(e)$ in residual network. We add edge e to this cut and will get minimum cut MAX in original network. Therefore, edge e can be crossing and is impossible to be never crossing.

It is impossible to be sometimes full and always crossing, sometimes full and sometimes crossing, never full and always crossing, never full and sometimes crossing.

Proof: These four cases can be proved together. No matter whether it is sometimes full or never full, the edge e which can be not full. We assume the max flow is MAX and $f(e) = m < c(e)$. We delete edge e and the max flow of residual network is $MAX - m$. There must exist minimum cut $MAX - m$ in residual network. If edge e can be crossing, then minimum cut in original network will be $MAX - m + c(e) > MAX$, which is impossible. Therefore edge e is never crossing. Always crossing and sometimes crossing are impossible.

It is possible to be sometimes full and never crossing, never full and never crossing.

Example:



The fat edge a has capcity 2, all other edges have capacity 1.

Figure 2:

In this network, b, c, d, f, g, h, i is sometimes full and a is never full. And a, b, c, d, f, g, h, i are all never crossing.