VE477 HW 6

Wu Jiayao 517370910257

November 14, 2019

1 Perfect matching in a bipartite graph

1.1 Proof

Suppose that matrix A has m rows and n columns. m = n. Denote $M_{i,j}^1$ the minor $M_{i,j}$ of A, $M_{i,j}^k$ as the minor $M_{i,j}$ of M^{k-1}

1. If the determinant is identically zero. Suppose there is a perfect matching, then for every row in A, there is only one element that has a value 1, denote as a_{i_x,j_x} for x^{th} row. The determinant of A is expressed as

$$\begin{aligned}
det(A) &= (-1)^{i_1+j_1} \cdot M_{i_1,j_1}^1 + 0 \times (n-1) \\
&= (-1)^{i_1+j_1+i_2+j_2} \cdot M_{i_2}^2 j_2 + 0 \times (n-2) \\
&= \cdots \\
&= (-1)^{i_1+j_1+\cdots+i_{m-1}+j_{m-1}} \cdot M_{i_{m-1},j_{m-1}}^{m-1} \\
&= (-1)^x \neq 0
\end{aligned}$$

   Since $M_{i_{m-1},j_{m-1}}^{m-1}$ is 1, as it represents in the perfect matching the match of the $(m-1)^{th}$ vertex in L, such leads to a contradiction. Hence there is no perfect matching.

2. If G has no perfect matching, there is at least one vertex in L that has no vertext in R to match, which means that there exists at least one row in A that is all 0. The determinant is identically zero in this condition.

1.2 Algorithm

Use the maximum matching algorithm described in the lecture.

Input : a bipartite graph $G\langle V, E\rangle$, $V = L \cup R$
Output: Whether it has a perfect matching

```
Add s and t
Connect s with all vertice in L with capacity 1
Connect t with all vertice in R with capacity 1
for each edge in G do
    Its capacity is 1
h = Ford - Fulkson(G)
if h = |R| then
    return true
return false
```

1.3 Complexity and error probability

The complexity is $\mathcal{O}(|V||E|)$. Since the existence of perfect matching matches the maximum of flows, the algorithm is always correct, with 0% of error probability.

1.4 Usefulness

It is correct. It has a resonable time complexity.

2 Critical Thinking

2.1 The middle node

Input : A singly linked list
Output: The middle node

$slow$ = the first node of the singly linked list
$fast = slow \to next$
while *$fast$ is not the last node of the singly linked list* **do**
    $slow = slow \to next$
    $fast = fast \to next \to next$
return $slow$

2.2 The cycle

Input : A singly linked list
Output: If there is a cycle

$slow$ = the first node of the singly linked list
$fast = slow \to next$
while *$fast$ is not the last node of the singly linked list* **do**
    if $fast == slow$ **then**
        return True
    $slow = slow \to next$
    $fast = fast \to next \to next$
return False

The complexity is $\mathcal{O}(n)$, where n is the length of the singly linked node.

3 The coupon collector desillusion

3.1 How many boxes

At least n.

3.2 Not finished

3.3 Expectation

$$E[X] = \sum_{i=1}^{n} \frac{n}{i} = n \sum_{i=1}^{n} \frac{1}{i} \approx n \int_{1}^{n} \frac{1}{x} dx = n \log n$$

It can be concluded that $E[X] = \Theta(n \log n)$

3.4 Coupon collector

The more boxes you have already owned, the more boxes you expect to buy (the bigger expectation) to collect a new boxes.