

# VE 477 Homework 1

Wang Yichao, ID: 517370910011

## • Exercise 1.

1. We know that the probability that a slot is chosen is  $1/n$  and  $1 - 1/n$  for not being chosen. Then if it is chosen times, there are  $\binom{n}{k}$  choices of  $k$  from  $n$ . Each of which has the possibility of  $(\frac{1}{n})^k (1 - \frac{1}{n})^{n-k}$ . Multiple them together and we can get the result.
2. (the probability of  $k$  keys for the most slot)  $<$  (the probability that at least one slot has  $k$  key)  $< nP_k$ .
3. We get  $n! \approx \sqrt{2\pi n} (\frac{n}{e})^n$  by Stirling's formula. Then we apply stirling for 3 times.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n^n}{k^k (n-k)^{n-k}} \cdot \frac{\sqrt{n}}{\sqrt{2\pi k(n-k)}}.$$

it is obvious that  $\frac{\sqrt{n}}{\sqrt{2\pi k(n-k)}} < 1/\sqrt{\pi} < 1$ .

So the LFH is smaller than  $\frac{n^{n-k}}{k^k (n-k)^{n-k}}$ .

$$\frac{n^{n-k}}{k^k (n-k)^{n-k}} = \frac{1}{k^k} \cdot \frac{n}{n-k}$$

Also notice that

$$\begin{aligned} &= \frac{1}{k^k} \cdot \left(1 + \frac{k}{n-k}\right)^{n-k} \\ &= \frac{1}{k^k} e^k \end{aligned}$$

Thus LHS is smaller than RHS. Qed.

4. optional

$$5. E[M] = \sum_{i=0}^n i \cdot P(M=i) \leq (k-1) \sum_{i=1}^{k-1} P_i + \sum_{i=k}^n i \cdot P_i,$$

where equality can not be reached.

choose  $k$ , we get  $E[M] \leq \frac{c \log n}{\log \log n} P\left(M \leq \frac{c \log n}{\log \log n}\right) + nP\left(M > \frac{c \log n}{\log \log n}\right)$

By Ex1.4, we can get  $P\left(M > \frac{c \log n}{\log \log n}\right) < \frac{1}{n^2}$ , plug in and we can get

$$\begin{aligned} E[M] &\leq \frac{c \log n}{\log \log n} P\left(M \leq \frac{c \log n}{\log \log n}\right) + \sum_{i=k}^n \frac{1}{n} \\ &\leq \frac{c \log n}{\log \log n} \cdot 1 + 1 \end{aligned}$$

$$\text{Thus } E(M) = \mathcal{O}\left(\frac{\log n}{\log \log n}\right).$$

## • Exercise 2.

---

### Algorithm 1 $MST_{renew}$

---

**Input:** graph  $G$ , MST  $T$ , and a edge  $e$  with weight decreased from  $w$  to  $w'$

**Output:** new MST  $T'$

- 1:  $T' \leftarrow T + e$
  - 2: Find the loop  $L$  that has the new edge (using BFS)
  - 3: linear search to get the edge  $e'$  with largest weight along  $L$
  - 4:  $T' \leftarrow T' - e'$
  - 5: **return**  $T'$
- 

Notice that there needs some proof to verify the algo.

## • Exercise 3.

1. Just follow the way we do addition in high school, from the LSD to MSD. The highest bit in the  $n+1$  array can be used as a carry bit.

2.(b) based on the fact that  $\text{mult}(x, y) = \text{mult}(2x, \lfloor y/2 \rfloor) + x \cdot (y - \lfloor y/2 \rfloor * 2)$ . it is obvious from case analysis when  $y$  is even and odd. binary multiplication is convenient for computer. When  $x = 0$  or  $y = 0$ , it is obvious. When  $y$  is odd,  $\text{mult}(x, y) = \text{mult}(2x, (y - 1)/2) + x$ , when  $y$  is even,  $\text{mult}(x, y) = \text{mult}(2x, y/2)$ .

Notice that the value of the second entry of  $\text{mult}$  is strictly decreasing, so the recursion will end.

(a) algo itself.

---

**Algorithm 2**  $\text{mult}$

---

**Input:** Two integers  $x, y$

**Output:** multiplication

```

1: function  $\text{MULT}(x, y)$ 
2:   if  $x = 0$  or  $y = 0$  then
3:     return 0
4:   else
5:     if  $y \equiv 0(\text{mod } 2)$  then
6:       return  $\text{mult}(x * 2, y/2)$ 
7:     else
8:       return  $\text{mult}(x * 2, (y - 1)/2) + x$ 
9:     end if
10:  end if
11: end function
```

---

• **Exercise 4.**

If no tie allowed. Since the fourth place out of a group of 5 can never be top 3 among 25 horses. Arrange 5 races that traverse all 25 horses. We can get 5 groups of local ranking, then take the first, second and third of each group. One more test among five local first horses and we can get the first among 25. Now we can get a table of 9 horses. To get the second and third of the 25 horses, we can have a last race between  $a_{12}, a_{13}, a_{21}, a_{22}, a_{31}$  (the upper left triangle without first of 25). Then we can tell the second and third. Seven races are needed.

I think it is beyond the scope of this course to prove that 6 tests are not enough.(we are not qualified for CMO!).

Also the condition is not complete, if ties are allowed, 5 tests will be enough if the fastest 3 horses are in same speed.

• **Exercise 5.**

1. Neither works. Greedy does not work for this NP. Counter example: assume the weight set are 1, 2, 3, 4, 5, 6 and the value set are 0, 0, 10, 10, 0, 0. Assume  $n = 7$ , using the first greedy algo, we will get items with weight 6 and 1, total value is 0. Second greedy algo will get items with weight 1, 2, and 4, total value is 10. Optimal selection is items with weight 3 and 4, with total value 20.

2. it is obvious that prime must be chosen, otherwise some space is wasted by module calculation. not too close from a power of 2 because in computer we use binary representation and the pattern of hash will be bad to organize if  $m$  is close to that.

3. Two examples in Ex5.1 is great. another example can be Set cover problem, Choosing the set that cover the most number of element is a bad greedy algorithm. Say the whole set is  $S = \{1, 2, 3, 4, 5, 6\}$ , there are subset  $S_1 = \{1, 2, 3, 4\}$ ,  $S_2 = \{1, 2, 3, 5\}$ ,  $S_3 = \{1, 2, 3, 6\}$ ,  $S_4 = \{5, 6\}$ . Greedy leads to  $S_1, S_2, S_3$ , while global optimal is  $S_1, S_4$ .

To explore more, please read *The Design of Approximate Algorithm* written by Williamson and Shmoys.