VE477 HW1

Wu Jiayao 517370910257

September 2019

1 EX.1

1.1

The probability is calculated as

$$
\begin{aligned}
P &= \frac{\binom{n}{k} 1^k \cdot (n-1)^{n-k}}{n^k} \\
&= (\frac{1}{n})^k (1-\frac{1}{n})^{n-k} \binom{n}{k}
\end{aligned}
$$

1.2

The probability of exactly one slot having k hash keys is nP_k. If $k > \frac{1}{2}n$, there will only one slot that has k hash keys. So the probability $P_k^{'} = nP_k$. If $k \leq \frac{1}{2}n$, there may be more than one slots that have k hash keys. To ensure all of the slots with most hash keys have k hash keys, the probability $P_k^{'} \leq nP_k$.
Overall, $P_k^{'} \leq nP_k$.

1.3

First is to prove that $\binom{n}{x}(\frac{k}{n})^x(1-\frac{k}{n})^{n-x} < \frac{k^x}{x!}e^{-k}$

$$
\begin{aligned}
f(x) &= \binom{n}{x}(\frac{k}{n})^x(1-\frac{k}{n})^{n-x} \\
&= \frac{n(n-1)...(n-x+1)}{x!} \cdot \frac{k^x}{n^x}(1-\frac{k}{n})^{n-x} \\
&= \frac{k^x}{x!} \cdot \frac{n(n-1)...(n-x+1)}{n^x} \cdot (1+\frac{1}{-\frac{n}{k}})^{(-\frac{n}{k})(-k+\frac{xk}{n})} \\
&< \frac{k^x}{x!} \cdot 1 \cdot e^{-k}
\end{aligned}
$$

For this problem,

$$
\begin{aligned}
P_k &= (\frac{1}{n})^k (1-\frac{1}{n})^{n-k} \binom{n}{k} \\
&< \frac{e^{-1}}{k!}
\end{aligned}
$$

According to Stirling formula,

$$n! \approx \sqrt{2\pi n}(\frac{n}{e})^n$$

which means,

$$\begin{aligned} P_k < \frac{e^{-1}}{k!} &\approx \frac{e^{-1}}{\sqrt{2\pi k}(\frac{k}{e})^k} \\ &= \frac{e^k}{k^k} \cdot \frac{e^{-1}}{\sqrt{2\pi k}} \\ &< \frac{e^k}{k^k} \end{aligned}$$

1.4

Skipped

1.5

$$\begin{aligned} E[M] &= \sum_{i=0}^{n} i \cdot P[M = i] \\ &\leq (k-1)\sum_{i=1}^{k-1} P_i + n\sum_{i=k}^{n} P_i \end{aligned}$$

Select a k that $k > \frac{c\log n}{\log\log n}$ and $k - 1 \leq \frac{c\log n}{\log\log n}$. We can conclude that

$$\begin{aligned} E[M] &\leq (k-1)\sum_{i=1}^{k-1} P_i + n\sum_{i=k}^{n} P_i \\ &\leq \frac{c\log n}{\log\log n} P(M \leq \frac{c\log n}{\log\log n}) + nP(M > \frac{c\log n}{\log\log n}) \end{aligned}$$

According to the conclusion in 1.4, $P(M > \frac{c\log n}{\log\log n}) < \frac{1}{n^2}$, therefore

$$\begin{aligned} E[M] &\leq \frac{c\log n}{\log\log n} P(M \leq \frac{c\log n}{\log\log n}) + n\sum_{i=k}^{n} \frac{1}{n^2} \\ &\leq \frac{c\log n}{\log\log n} \cdot 1 + 1 \end{aligned}$$

As a conclusion,

$$E[M] = \mathcal{O}(\frac{\log n}{\log\log n})$$

2 EX.2

Algorithm 1: Determine the minimum spanning tree

input : the decreased edge E(u,v), minimum spanning tree T of graph G
output: new minimum spanning tree T'

```
T' ← T + E(u,v)
L_0 ← The only cycle in T' that is formed due to E(u,v)
w_0 ← weight of E(u,v)
E' ← The edge in L_0 that has the largest weight
T' ← T' − E'
return T'
```

3 EX.3

3.1

Skipped.

3.2

3.2.1

Algorithm 2: mult(x,y)

input : two integers to multiply x,y
output: result of multiply n

```
if x=0 or y=0 then
    return 0
return mult(2x, ⌊y/2⌋) + x × (y mod 2)
```

3.2.2

It is to prove that when mult(x,y) is right for integers $(2m, n)$, which means $mult(2m, n) = 2mn$, then mult(x,y) is also correct for $(m, 2n)$,$(m, 2n + 1)$.

First, whether $m = 0$ or $n = 0$, the statement is proved true.

Then

$$mult(m, 2n) = mult(2m, n) + 2m * 0 = 2mn = m \times 2n$$

$$mult(m, 2n + 1) = mult(2m, n) + 2m * 1 = 2mn + 2m = m \times (2n + 1)$$

Hence the statement is proved true. The correctness of the algorithm is proved.

4 EX.4

The minimum number is 7.

Divide the 25 horses into 5 groups of 5. Each group hold one race. Record the first, second, third fastest, noted as h_{11}, h_{12}, h_{13} for the first, second, third of group 1, for example. Current race number is 5.

Hold a race between $h_{11}, h_{21}, h_{31}, h_{41}, h_{51}$, record the first, second,third as h_{61}, h_{62}, h_{63}. h_{61} is the fastest. Current race number is 6.

Suppose that h_{62} is h_{i1} and h_{63} is h_{j1}. Hold a race between h_{62},h_{63},h_{i2},h_{i3},h_{j2}. Record the first, second, third as h_{71}, h_{72}, h_{73}. h_{71} is the second fastest. h_{72} is the third fastest.Current number is 7.

5 EX.5

5.1

Neither will solve the problem. A counterexample is to get $n = 8$ from $S = \{2, 3, 5, 7, 9\}$ where the solution is $\{3, 5\}$

5.2

Skipped.

5.3

Problem: Use at least as possible coins to give out a combination of $51. You have coins with value $40, $25, $7, $2, $1.

According to greedy algorithm, locally optimal is given by $40,$7,$2,$2, four coins.

But actually the globally optimal is given by $25,$25,$1, three coins.