算法设计与分析: 作业 #4

201928015059003
杨天健

2019 年 10 月 28 日

Question 1

Sort service time in ascending order, and then offer them service one by one. In this way, the total wating time will be minimized.

First, we sort these people by service time in ascending order, say $a_1, a_2, ..., a_n$. Let $w[k]$ to be the optimal total waiting time if we only have person $a_1, a_2, ..., a_k$, which has top-k service time. Obviously, $w[1] = t_{a_1}$. Then, consider inserting a_{k+1} at the back of a_p, then:

$$w[k+1] = \min_{p=1,2,...,k} \underbrace{w[k]}_{subproblem} + \underbrace{\sum_{j=1}^{p} t_{a_j}}_{waitforprevious} + \underbrace{(k-p) \cdot t_{a_{k+1}}}_{people\, after\, a_{k+1}\, have\, to\, wait\, for\, a_{k+1}} + \underbrace{t_{a_{k+1}}}_{itself} \tag{1}$$

Because $t_{a_{k+1}} \geq t_{a_j}$ for all $j = 1, 2, ..., n$, the equation above has a lower bound:

$$\geq \min_{p=1,2,...,k} w[k] + \sum_{j=1}^{p} t_{a_j} + \sum_{j=p+1}^{k} t_{a_j} + t_{a_{k+1}} \tag{2}$$

$$= w[k] + \sum_{j=1}^{k} t_{a_j} + t_{a_{k+1}} \tag{3}$$

Consider we just append a_{k+1} in the last place. This lower bound can be attained. Therefore, by induction, we can use the greedy strategy stated above to solve the problem correctly.

Question 2

By greedy method that Huffman applies, the coding of a - g are listed below:

letter	code
a	111111
b	111110
c	11110
d	1110
e	110
f	10
g	0

Intuitvely, if we have n letters $a_1, a_2, ...a_n$ whose frequency match fibonacci sequence, the code of a_i will be:

$$code(a_i) = \begin{cases} \underbrace{11...1}_{\times(n-1)} & (i = 1) \\ \underbrace{11...1}_{\times(n-i)} 0 & (i = 2, 3, ..., n) \end{cases} \tag{4}$$

Question 3

(a)

Every feasible solution can be constrcted by permuting these n tapes and then fetch the top-k.

Assume that we have a arbitary permutation $S \equiv a_{s_1}, a_{s_2}, ..., a_{s_n}$. $\sum_{i=1}^{k} a_{s_i} \leq L$ and $\sum_{i=1}^{k+1} a_{s_i} > L$.

Then we propose that we fetch the tape in ascending order by length, say $T \equiv a_{t_1}, a_{t_2}, ..., a_{t_n}$. Then $\sum_{i=1}^{k} a_{t_i} \leq \sum_{i=1}^{k} a_{s_i} \leq L$. For each possible permutation S, we have a k. And the same k is legal if we fetch tapes in ascending order. Say, this greedy method will obtain a result that is not worse than any feasible solutions. That means we can obtain an optimal solution using greedy method.

(b)

The usage of the tape can be 0 if $\min_{a_1, a_2, ..., a_n} > L$.

(c)

If $a_1 = 1, a_2 = 2, a_3 = 3$ and $L = 4$, choosing a_1 and a_3 can use the full tape. But the usage drop to $\frac{3}{4}$ if we apply greedy strategy.

Question 6

We define matroid of MST algorithm as $M = (S, L)$. Here, $S = E$ and $L = e|G(V, e) without loops$. G is the graph provided and V is the set of vertices.

We first prove heredity. For $A \in L$ and $B \subseteq A$. Obviously, if there are no loops in $G_A(V, A)$, there must be no loops in $G_B(V, B)$.

Next, we prove exchage property. For $A \in L$ and $B \in L$ and $|A| < |B|$ (here, $|S|$ is the cardinality of S), their connected components are $|V| - |A|$ and $|V| - |B|$. So $|V| - |A| \geq |V| - |B|$. So we can know that there must be an edge $e_0 \in B$ in G_B that connects **two components** in G_A, meaning that $G(V, A \cup x)$ has no loops.

Because we find MST, the weight function on $e \in E$ can be $f(e) = -w(e)$, here w is the weight of edge. Therefore, minimizing the weight of MST equals to maximizing the function f.

By using kruskal algorithm, we continuously add $e \in S$ that has maximum weight function, which is the same procedure as the general greedy algorithm defined based on matroid theory. Therefore, this algorithm is right.

To prove prim algorithm is right, we just have to show that the edges selected in kruskal algorithm can be also selected by prim.

Assume that we have an edge $e \in E$ that $f(e)$ is bigger that any of the nodes in a candidate tree T, we can add E, then there must be a loop in T. Then, we delete another edge $e_0 \in T$ which has minimal weight function f, we will construct another tree T' that is better.

Therefore, the edges that connects different components which has maximum f (another word, minimum w) must be added to construct MST, which is the same as kruskal.