

CS60007 Algorithm Design and Analysis 2018

Assignment 1

Palash Dey and Swagato Sanyal
Indian Institute of Technology, Kharagpur

Please submit the solutions of the problems 2, 6, 11, 12 and 13 (written in red color). You are strongly encouraged to do all the problems for your own practice. The deadline is August 13, 2018.

This is a preliminary version and may contain errors. Please send an email to the instructors if you find any error. Thank you for your cooperation.

1. There is a sequence of n activities a_1, a_2, \dots, a_n with corresponding utilities u_1, u_2, \dots, u_n . You wish to perform all these n activities according to this sequence within k days. If you perform the activities from a_i to a_j for some $1 \leq i \leq j \leq n$ on the j -th day, then your utility U_j for the j -th day is $\max\{u_\ell : i \leq \ell \leq j\}$. Your total utility is $\sum_{\ell=1}^k U_\ell$. Design a greedy algorithm to find the sequence of activities you will perform on every day which maximizes your total utility.
2. ~~[15 marks] There are n jobs J_1, J_2, \dots, J_n with duration $a_1, a_2, \dots, a_n \in \mathbb{N}^+$ and deadlines $d_1, d_2, \dots, d_n \in \mathbb{N}$ respectively. All the jobs are available at the beginning (that is at time 0). If a job J_ℓ is completed at time t_ℓ with $t_\ell > d_\ell$ then the utility u_ℓ incurred for job J_ℓ is -1 (which is a penalty); if $t_\ell \leq d_\ell$, then $u_\ell = 1$ (which is a profit). You have only one machine. So, you cannot process more than one jobs simultaneously. Every job, once started, must run till it finishes. Design a greedy algorithm to find the schedule for these n jobs which maximizes $\sum_{\ell=1}^n u_\ell$.~~
3. We are given a decimal string $s = (a_1, a_2, \dots, a_n)$ of n digits (that is $a_i \in \mathbb{N}$ and $0 \leq a_i \leq 9$) and a positive integer k . Design an algorithm for find the lexicographic minimum string which can be obtained from s by performing at most k swaps. Note that we are allowed to swap consecutive elements only.
4. Suppose you have to give Re N to your friend. You have enough number of 500, 200, 100, 50, 20, 10 rupee notes each at your disposal. Your goal is to give Re N to your friend with minimum number of notes. For example, Re 600 can be changed

using 3 Re 200 notes as well as using 1 Re 500 note and 1 Re 100 note. However, the later one uses minimum number of notes.

- ▷ Either prove correctness or provide counter example of the following greedy strategy: keep picking highest denomination as much as you can!
 - ▷ Provide a set of denominations for which the above greedy strategy will fail.
5. An independent set of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a subset $\mathcal{W} \subseteq \mathcal{V}$ such that there does not exist any edge $e = \{u, v\} \in \mathcal{E}$ with both end points in \mathcal{W} (that is, $|e \cap \mathcal{W}| \leq 1$). An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is called an interval graph if every vertex $v \in \mathcal{V}$ can be associated with some interval $\mathcal{J}_v = [a, b]$ in \mathbb{R} such that there is an edge e between u and v if and only if $\mathcal{J}_u \cap \mathcal{J}_v \neq \emptyset$. Let \mathcal{G} be an interval graph and $\{\mathcal{J}_v : v \in \mathcal{V}\}$ be the intervals associated with its vertices. Then design an algorithm for computing the size of a maximum independent set of \mathcal{G} .
 6. **[15 marks]** Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected, weighted graph. Let \mathcal{T} and \mathcal{T}' be two MSTs of \mathcal{G} and $\alpha \in \mathbb{R}$. Then the number of edges in \mathcal{T} of weight α is the same as the number of edges in \mathcal{T}' of weight α .
 7. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected, weighted graph, $v \in \mathcal{V}$ be any vertex, and e be an edge with minimum weight among all the edges that incident on v . Prove that there exists a MST which includes the edge e .
 8. Let \mathcal{G} be a connected, weighted graph. Prove that, if all edge weights in \mathcal{G} are distinct, then \mathcal{G} has exactly one MST.
 9. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected, weighted graph. Let $\mathcal{S} \subset \mathcal{V}$ with $\mathcal{S} \neq \emptyset$, $\mathcal{S} \neq \mathcal{V}$, and $e' = \operatorname{argmin}_{e \in \mathcal{E} : e = \{u, v\}, u \in \mathcal{S}, v \in \mathcal{V} \setminus \mathcal{S}} \{w(e)\}$. Then there exists a MST T where $e' \in T$.
 10. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected, weighted graph and e be an edge with minimum weight in \mathcal{G} . Then prove that there exists a MST which includes the edge e .
 11. **[Independent Set of Binary Tree]** Refer to the definition of Independent Set of an undirected graph in problem 5. In this exercise you will compute a maximum weighted independent set of a binary tree. Your algorithm will be given as input a pointer to the root of a binary tree. Each node v has a *weight* $w(v)$ and two pointers $\text{left}(v)$ and $\text{right}(v)$ leading to the left and right children of v respectively; if a child is absent, the corresponding pointer is NULL. The weight of an independent set in a tree is the sum of the weights of the nodes in it. A maximum weight independent set is an independent set whose weight is maximum.
 - (a) **[4 marks]** Derive a recursive relation for the weights of a maximum weight independent set of various subtrees.
 - (b) **[4 marks]** Design an efficient algorithm to compute the weight of a maximum weight independent set of the input tree.

- (c) **[4 marks]** Modify your algorithm to also compute a maximum weight independent set of the input tree, in addition to its weight.
- (d) **[3 marks]** Discuss running times of your algorithms.
12. Suppose a trader has a container vehicle capable of shipping at most W amount of goods (W is an integer). Suppose that he has n items $\{1, \dots, n\}$. Each item i has a price v_i and an integral weight w_i . The trader wants to ship a set $S \subseteq \{1, \dots, n\}$ of items of highest possible total price subject to the constraint that the total weight of the items is at most W .
- (a) **[1.5 marks]** Mathematically formulate the constraint of the trader.
- (b) **[1.5 marks]** Mathematically formulate the objective that the trader wants to maximize subject to his constraint.
- (c) **[4 marks]** For an integer \mathcal{W} , let $c(i, \mathcal{W})$ be the maximum total price of a set of items in $\{1, \dots, i\}$ with total weight at most \mathcal{W} . Derive a recursive relation for $c(i, \mathcal{W})$.
- (d) **[4 marks]** Design an efficient algorithm for computing $c(n, W)$.
- (e) **[3 marks]** Modify your algorithm to also find a set of items that correspond to $c(n, W)$.
- (f) **[1 marks]** Discuss running times of your algorithms.
13. Consider the following properties of an undirected graph G with n vertices.
- (a) G is connected.
- (b) G is acyclic.
- (c) G has $n - 1$ edges.

Prove that if G satisfies any two of the above three properties, then G also satisfies the third property, i.e.,

- (i) **[2 marks]** if G satisfies properties (a) and (b), then G satisfies property (c),
- (ii) **[2 marks]** if G satisfies properties (a) and (c), then G satisfies property (b),
- (iii) **[2 marks]** if G satisfies properties (b) and (c), then G satisfies property (a).

As you must know, graphs satisfying these three properties are called trees.

14. We are given two English strings s_1 and s_2 of length n each. Design a dynamic programming based algorithm to find the minimum number of the following operations that one requires to perform on s_1 to convert it into s_2 .
- ▷ Insert any symbol at any position of the string.

- ▷ Delete any symbol from the string.
 - ▷ Replace any symbol in the string by another symbol.
15. A string s is called a sub-sequence of another string t if s can be obtained from t by deleting some symbols from s . Design a dynamic programming based algorithm which finds a longest common sub-sequence of two input strings.
 16. Design a dynamic programming based algorithm which finds a subset of a set of integers (given as an array of integers as input) which has maximum sum of its elements.
 17. Design a dynamic programming based algorithm for problem 4 which works for any set of denominations.
 18. There are n jobs J_1, J_2, \dots, J_n with start and end times $s_1, s_2, \dots, s_n \in \mathbb{N}$ and deadlines $t_1, t_2, \dots, t_n \in \mathbb{N}$ respectively with $t_\ell > s_\ell$ for every $\ell \in [n]$. The weight of the job J_ℓ is $w_\ell \in \mathbb{N}$ for $\ell \in [n]$. You have only one machine. So, you cannot process more than one jobs simultaneously. Every job, once started, must run till it finishes. Design a dynamic programming based algorithm for the problem of scheduling a subset $\mathcal{S} \subseteq \{J_\ell : \ell \in [n]\}$ of jobs which maximizes $\sum_{J_\ell \in \mathcal{S}} w_\ell$.
 19. Suppose we have a stick of length $n \in \mathbb{N}^+$. Design a dynamic programming based algorithm to break the stick into at least 2 pieces of integral lengths which maximizes the product of the lengths of the pieces.
 20. Suppose we have a stick of length $n \in \mathbb{N}^+$. Let $p : \mathbb{N}^+ \rightarrow \mathbb{N}^+$ be a function which maps every positive integer ℓ to the utility of a stick of length ℓ . Design a dynamic programming based algorithm to break the stick into pieces which maximizes the total utility (the sum of utilities of the pieces).