# Algorithms Tutorial Problems 3
# Greedy Strategy Problems

1. There are N robbers who have stolen N items. You would like to distribute the items among the robbers (one item per robber). You know the precise value of each item. Each robber has a particular range of values they would like their item to be worth (too cheap and they will not have made money, too expensive and they will draw a lot of attention). Devise an algorithm that either distributes the items so that each robber is happy or determines that there is no such distribution.

2. (Timing Problem in VLSI chips) Consider a complete binary tree with $n = 2^k$ leaves. Each edge has an associated positive number that we call the length of the edge (see figure below). The distance from the root to a leaf is the sum of the lengths of all edges from the root to the leaf. The root emits a clock signal and the signal propagates along all edges and reaches each leaf in time proportional to the distance from the root to that leaf. Design an algorithm which increases the lengths of some of the edges in the tree in a way that ensures that the signal reaches all the leaves at the same time while the total sum of the lengths of all edges is minimal. (For example, in the picture below if the tree A is transformed into trees B and C all leaves of B and C have a distance of 5 from the root and thus receive the clock signal at the same time, but the sum of the lengths of the edges in C is 17 while sum of the lengths of the edges in B is only 15.)
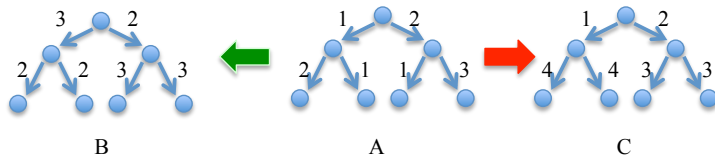
3. Assume that you are given a complete graph G with weighted edges such that all weights are distinct. We now obtain another complete weighted graph G' by replacing all weights $w(i, j)$ of edges $e(i, j)$ with new weights $w(i, j)^2$.

    (a) Assume that T is the minimal spanning tree of G. Does T necessarily remain the minimal spanning tree for the new graph G'?

    (b) Assume that p is the shortest path from a vertex u to a vertex v in G. Does p necessarily remain the shortest path from u to v in the new graph G'?

4. Assume that you are given a complete weighted graph G with n vertices $v_1, \ldots, v_n$ and with the weights of all edges distinct and positive. Assume that you are also given the minimum spanning tree T for G. You are now given a new vertex v_{n+1} and the weights $w(n+1, j)$ of all new edges $e(n+1, j)$ between the new vertex v_{n+1} and all old vertices $v_j \in G$, $1 \le j \le n$. Design an algorithm which produces a minimum spanning tree T' for the new graph containing the additional vertex v_{n+1} and which runs in time $O(n \log n)$.

5. You have n items for sale, numbered from 1 to n. Alice is willing to pay $a[i] > 0$ dollars for item i, and Bob is willing to pay $b[i] > 0$ dollars for item i. Alice is willing to buy no more than A of your items, and Bob is willing to buy no more than B of your items. Additionally, you must sell each item to either Alice or Bob, but not both, so you may assume that $n \le A+B$. Given $n, A, B, a[1 \ldots n]$ and $b[1 \ldots n]$, you have to determine the **maximum** total amount of money you can earn in $O(n \log n)$ time.

6. Assume that you have an unlimited number of \$2, \$1, 50c, 20c, 10c and 5c coins to pay for your lunch. Design an algorithm that, given the cost that is a multiple of 5c, makes that amount using a minimal number of coins.

7. Assume that the denominations of your $n + 1$ coins are $1, c, c^2, c^3, \ldots, c^n$ for some integer $c > 1$. Design a greedy algorithm which, given any amount, makes that amount using a minimal number of coins.

8. Give an example of a set of denominations containing the single cent coin for which the greedy algorithm does not always produce an optimal solution.

9. Given two sequences of letters A and B, find if B is a subsequence of A in the sense that one can delete some letters from A and obtain the sequence B.

10. There is a line of 111 stalls, some of which lack a roof and need to be covered with boards. You can use up to 11 boards, each of which may cover any number of consecutive stalls. Cover all the necessary stalls, while covering as few total stalls as possible.

11. Let X be a set of n intervals on the real line. A subset of intervals $Y \subseteq X$ is called a tiling path if the intervals in Y cover the intervals in X, that is, any real value that is contained in some interval in X is also contained in some interval in Y. The size of a tiling cover is just the number of intervals. Describe and analyse an algorithm to compute the smallest tiling path of X as quickly as possible. Assume that your input consists of two arrays $X_L[1..n]$ and $X_R[1..n]$, representing the left and right endpoints of the intervals in X.
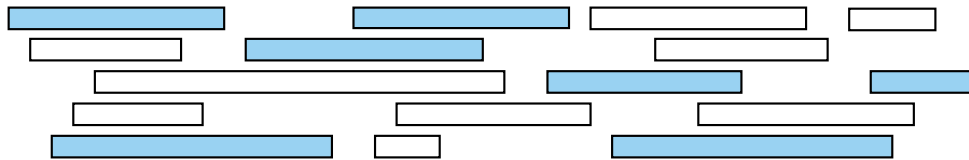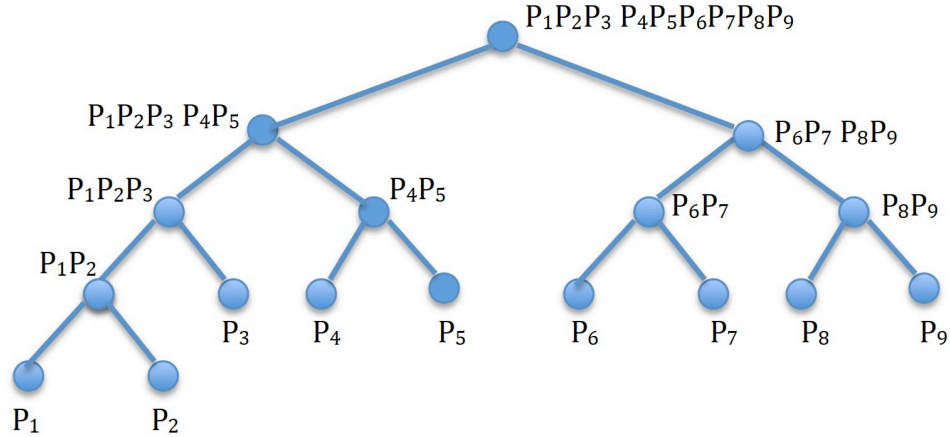

A set of intervals. The seven shaded intervals form a tiling path.

12. Assume that you are given n white and n black dots lying in a random configuration on a straight line, equally spaced. Design a greedy algorithm which connects each black dot with a (different) white dot, so that the total length of wires used to form such connected pairs is minimal. The length of wire used to connect two dots is equal to the straight-line distance between them.
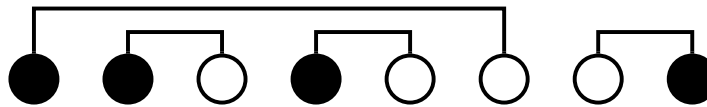
13. Assume you are given n sorted arrays P_i, $1 \le i \le n$, of different sizes e_i. You are allowed to merge any two arrays into a single new sorted array and proceed in this manner until only one array is left. Design an algorithm that achieves this task and uses minimal total number of moves of elements of the arrays. Give an informal justification for why your algorithm is optimal.

$P_1P_2P_3\ P_4P_5P_6P_7P_8P_9$

$P_1P_2P_3\ P_4P_5$

$P_6P_7\ P_8P_9$

$P_1P_2P_3$

P_4P_5

P_6P_7

P_8P_9

P_1P_2

P_3 P_4 P_5 P_6 P_7 P_8 P_9

P_1 P_2

14. A photocopying service with a single large photocopying machine faces the following scheduling problem. Each morning they get a set of jobs from customers. They want to schedule the jobs on their single machine in an order that keeps their customers the happiest. Customer i's job will take t_i time to complete. Given a schedule (i.e., an ordering of the jobs), let C_i denote the finishing time of job i. For example, if job i is the first to be done we would have $C_i = t_i$, and if job j is done right after job i, we would have $C_j = C_i + t_j$. Each customer i also has a given weight w_i which represents his or her importance to the business. The happiness of customer i is expected to be dependent on the finishing time of their job. So the company decides that they want to order the jobs to minimise the weighted sum of the completion times, $\sum_{i=1}^n w_i C_i$. Design an efficient algorithm to solve this problem. That is, you are given a set of n jobs with a processing time t_i and a weight w_i for job i. You want to order the jobs so as to minimise the weighted sum of the completion times, $\sum_{i=1}^n w_i C_i$.

15. You are running a small manufacturing shop with plenty of workers, but just a single milling machine. You have to produce n items; item i requires m_i machining time first, then p_i polishing time by hand; finally it is packaged by hand and this takes c_i amount of time. The machine can mill only one item at a time, but your workers can polish and package in parallel as many objects as you wish. You have to determine the order in which the objects should be machined so that the whole production is finished as quickly as possible. Prove that your solution is optimal.

16. You have a processor that can operate 24 hours a day, every day. People submit requests to run daily jobs on the processor. Each such job comes with a start

time and an end time; if a job is accepted, it must run continuously, every day, for the period between its start and end times. (Note that certain jobs can begin before midnight and end after midnight; this makes for a type of situation different from what we saw in the activity selection problem.) Given a list of n such jobs, your goal is to accept as many jobs as possible (regardless of their length), subject to the constraint that the processor can run at most one job at any given point in time. Provide an algorithm to do this with a running time that is polynomial in n. You may assume for simplicity that no two jobs have the same start or end times.

17. Assume that you got a fabulous job and you wish to repay your student loan as quickly as possible. Unfortunately, the bank *Western Road Robbery* which gave you the loan has the condition that you must start your monthly repayments by paying off \$1 and then each subsequent month you must pay either double the amount you paid the previous month, the same amount as the previous month or half the amount you paid the previous month. On top of these conditions, your schedule must be such that the last payment is \$1. Design an algorithm which, given the size of your loan, produces a payment schedule which minimises the number of months it will take you to repay your loan while satisfying all of the bank's requirements. If the optimality of your solution is obvious from the algorithm description, you do not have to provide any further correctness proof.

18. Alice wants to throw a party and is deciding whom to invite. She has n people to choose from, and she has created a list consisting of pairs of people who know each other. She wants to invite as many people as possible, subject to two constraints: at the party, each person should have at least five other people whom they know and at least five other people whom they do not know. Give an efficient algorithm that takes as input the list of n people and the list of all pairs who know each other and outputs a subset of these n people which satisfies the constraints and which has the largest number of invitees. Argue that your algorithm indeed produces a subset with the largest possible number of invitees.

19. In Elbonia cities are connected with one way roads and it takes one whole day to travel between any two cities. Thus, if you need to reach a city and there is no direct road, you have to spend a night in a hotel in all intermediate cities. You are given a map of Elbonia with toll charges for all roads and the prices of the cheapest hotels in each city. You have to travel from the capital city C

to a resort city R. Design an algorithm which produces the cheapest route to get from C to R.

20. You are given a set S of n overlapping arcs of the unit circle. The arcs can be of different lengths. Find a largest subset P of these arcs such that no two arcs in P overlap (largest in terms of the total number of arcs, not in terms of the total length of these arcs). Prove that your solution is optimal.

    Hint: compare with problem 16.

21. You are given a set S of n overlapping arcs of the unit circle. The arcs can be of different lengths. You have to stab these arcs with a minimal number of needles so that every arc is stabbed at least once. In other words, you have to find a set of as few points on the unit circle as possible so that every arc contains at least one point. Prove that your solution is optimal.

    Hint: compare with problem 16.

22. You are given a connected graph with weighted edges. Find a spanning tree such that the largest weight of all of its edges is as small as possible.

23. You need to write a very long paper titled "The Meaning of Life". You have compiled a sequence of books in the order that you will need them, some of them multiple times. Such a sequence might look something like this:

$$B_1, B_2, B_1, B_3, B_4, B_5, B_2, B_6, B_4, B_1, B_7, \ldots$$

Unfortunately, the library only lets you borrow ten books at a time, so every now and then you have to make a trip to the library to exchange books. On each trip you can exchange any number of books. Design an algorithm which decides which books to exchange on each library trip so that the total number of trips which you will have to make to the library is as small as possible.

24. Assume that you are given a set X of n disjoint intervals $I_1, I_2, \ldots, I_n$ on the real line and a number $k \leq n$. You have to design an algorithm which produces a set Y of at most k intervals $J_1, J_2, \ldots, J_k$ which cover all intervals from X (i.e. $\bigcup_{i=1}^{n} I_i \subseteq \bigcup_{j=1}^{k} J_j$) and such that the total length of all intervals in Y is minimal. (Note that we do not require that $Y \subseteq X$, i.e., intervals J_j can be new. For example, if your set X consists of intervals $[1, 2]$, $[3, 4]$, $[8, 9]$, and $[10, 12]$ and if $k = 2$, then you should choose intervals $[1, 4]$ and $[8, 12]$ of total length $3 + 4 = 7$.)

25. Assume that you are given a complete undirected graph $G = (V, E)$ with edge weights $\{w(e) : e \in E\}$ and a proper subset of vertices $U \subset V$, $U \neq V$. Design an algorithm which produces the lightest spanning tree of G in which all nodes of U are leaves (there might be other leaves in this tree as well).

26. You are given a connected graph with weighted edges with all weights distinct. Prove that such a graph has a unique minimum spanning tree.

27. You have N students with varying skill levels and N jobs with varying skill requirements. You want to assign a different job to each student, but only if the student meets the job's skill requirement. Design an algorithm to determine the maximum number of jobs that you can successfully assign.

28. There are N towns situated along a straight line. The location of the i'th town is given by the distance of that town from the westernmost town, d_i (so the location of the westernmost town is 0). Police stations are to be built along the line such that the i'th town has a police station within A_i kilometers of it. Design an algorithm to determine the minimum number of police stations that would need to be built.

29. There are N people that you need to guide through a difficult mountain pass. Each person has a speed in days that it will take to guide them through the pass. You will guide people in groups of up to K people, but you can only move as fast as the slowest person in each group. Design an algorithm to determine the fewest number of days you will need to guide everyone through the pass.

30. There are N courses that you can take. Each course has a minimum required IQ, and will raise your IQ by a certain amount when you take it. Design an algorithm to find the minimum number of courses you will need to take to get an IQ of at least K.