

COMP 271 Design and Analysis of Algorithms
2003 Spring Semester
Question Bank 4 – Selected Solutions

There is some overlap between these question banks and the tutorials. Solving these questions (and those in the tutorials) will give you good practice for the midterm. Some of these questions (or similar ones) will definitely appear on your exam! Note that you do not have to submit answers to these questions for grading. Your TAs will discuss answers to selected questions in the tutorials.

1. (a) The Huffman code uses 1 bit for 'a', 2 bits for 'd', 3 bits for 'b' and 3 bits for 'c'. It therefore uses $1 \cdot 14 + 2 \cdot 10 + 3 \cdot 6 + 3 \cdot 3 = 61$ bits in total. The code using 2 bits per character uses $2 \cdot (14 + 10 + 6 + 3) = 66$ bits so the Huffman code is better.
(b) No. Suppose that we have four characters a, b, c, d with respective frequencies $\{10, 11, 12, 13\}$. In the optimal code every character is represented by 2 bits so every 2-bit fixed length code is optimal. For example $\{a = 00, b = 01, c = 10, d = 11\}$ and $\{a = 00, b = 10, c = 01, d = 11\}$ are both optimal codes.
(c) (i) $f_1 + f_2 \leq f_3 \leq \dots \leq f_n$
(ii) Take the leaf in T_2 corresponding to frequency $f_1 + f_2$ and replace it with an internal node having two leaf children, one with frequency f_1 and the other with frequency f_2 .
2. (solution from CLRS) The optimal strategy is the obvious greedy one. Starting with a full tank of gas, Professor Midas should go to the farthest gas station he can get to within n miles of Newark. Fill up there. Then go to the farthest gas station he can get to within n miles of where he filled up, and fill up there, and so on. Looked at another way, at each gas station, Professor Midas should check whether he can make it to the next gas station without stopping at this one. If he can, skip this one. If he cannot, then fill up.

Professor Midas doesn't need to know how much gas he has or how far the next station is to implement this approach, since at each fillup, he can determine which is the next station at which he needs to stop. This problem has optimal substructure. Suppose there are m possible gas stations. Consider an optimal solution with s stations and whose first stop is at the k th gas station. Then the rest of the optimal solution must be an optimal solution to the subproblem of the remaining $m - k$ stations. Otherwise, if there were a better solution to the subproblem, i.e., one with fewer than $s - 1$ stops, we could use it to come up with a solution with fewer than s stops for the full problem, contradicting our supposition of optimality. This problem also has the greedy-choice property. Suppose there are k gas stations beyond the start that are within n miles of the start. The greedy solution chooses the k th station as its first stop. No station beyond the k th works as a first stop,

since Professor Midas runs out of gas first. If a solution chooses a station $j < k$ as its first stop, then Professor Midas could choose the k th station instead, having at least as much gas when he leaves the k th station as if he had chosen the j 'th station. Therefore, he would get at least as far without filling up again if he had chosen the k th station. If there are m gas stations on the map, Midas needs to inspect each one just once. The running time is $O(m)$.

3. The greedy algorithm of choosing the leftmost point x in X starting a unit interval there, scanning to the right until we find the leftmost point in X not yet covered, starting a unit-interval there, etc. works. Since it scans all the points once it works in $O(n)$ time.

To see that this algorithm is correct we show that it has both the *greedy choice property* and the *optimal substructure property*.

This problem has the *greedy choice property*.

Let S be an optimal solution to the problem, i.e., a smallest collection of unit intervals that cover the points. Let I be the *leftmost* interval in S . Note that $x_1 \in I$ (since if it wasn't, I would cover no points and we could throw it away, reducing the size of S). So $I = [u, 1 + u]$ for some $u \leq x_1$. Since there are no points to the left of x_1 we find that $[x_1, 1 + x_1]$ covers all of the points in $X - I$ so $\{[x_1, 1 + x_1]\} \cup (S - I)$ covers all of the points in X using the same number of intervals as S . So there is an optimal solution containing $[x_1, 1 + x_1]$.

We also see that the problem has the *optimal substructure property*.

Suppose S is a solution to the original problem that uses $I = [x_1, 1 + x_1]$. Now $S - \{I\}$ must be an optimal solution to the problem of covering $X' = X - (X \cap I)$ (the points in X that are not in I) because. If it wasn't, then there would be a collection of intervals S' covering X' such that $|S'| < |S - \{I\}| = |S| - 1$. Combining S' and I would give a solution to X smaller than S , contradicting the optimality of S . Since every optimal solution covering X' has the same size, we can find an optimal solution covering S by combining I with any optimal cover of X' .

Combing the two parts above we see that we can get an optimal solution by starting a unit interval $I = [x_1, 1 + x_1]$ at x_1 , throwing away all of the points in I , and then starting again. This is exactly what our algorithm does so we are done.