Institut for Matematik og Datalogi
Syddansk Universitet, Odense

12. november 2017
JBJ

Eksaminatorier DM534

Husk at læse de relevante slides før du forsøger at løse en opgave.

Uge 46

1. Give an intuitive argument which shows that a graph $G = (V, E)$ is connected if and only if it has a spanning tree. Hint: how would you construct a spanning tree if the graph is connected?

2. Use the Greedy algorithm to find a minimum spanning tree in the graph from Figure 1.

3. Prove that if we add one new edge between two vertices u, v of a tree, then the resulting graph will have a cycle (comment: in fact one can show that it will have exactly one cycle).

4. Let $G = (V, E)$ be a connected graph with a cost function c on its edges and suppose that the edge e has the lowest cost, that is, for every edge $e' \in E$ we have $c(e) \leq c(e')$.

   - Prove that e is contained in some minimum spanning tree of G. Hint: use the result from the exercise above to show how to construct a minimum spanning tree containing e: Take a minimum spanning tree T and assume it does not contain e. Then look at the cycle we get by adding e to T.
   - Can you conclude even more if we know that $c(e) < c(e')$ for all $e' \in E$ with $e' \neq e$?

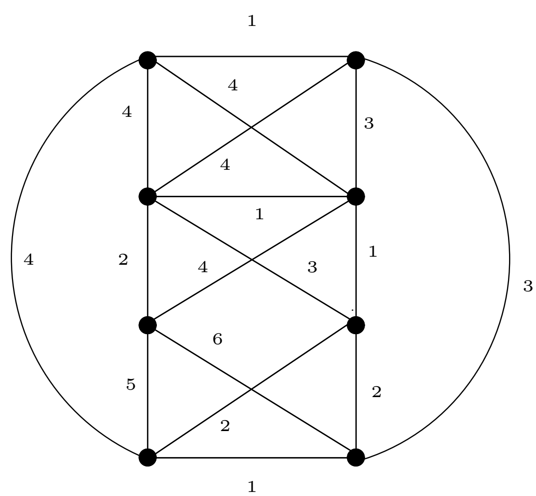

Figure 1: An edge weighted graph

5. Suppose $G = (V, E)$ is a graph with edge cost function c. Devise an algorithm to find a spanning tree of maximum cost. Suppose now that you have an algorithm $\mathcal{A}$ for minimum spanning tree and you are not allowed to build a new algorithm but you may define a new cost function c'. Show how to define c' so that we can solve the maximum cost spanning tree problem for G, c by using the algorithm $\mathcal{A}$ on G, c'. Hint: you may use the fact that every tree with n nodes has $n - 1$ edges (extra exercise: prove this), which in particular holds for every spanning tree on G if n is the number of vertices of G.

6. For which of the following coin systems S does the greedy algorithm (always use as many coins of largest value as possible) work? If it does not work, you should give an example where it uses too many coins. If it works, you should try to prove this.

   (a) $S = \{10, 7, 2, 1\}$
   (b) $S = \{10, 7, 5, 2, 1\}$
   (c) $S = \{14, 10, 7, 5, 2, 1\}$
   (d) $S = \{20, 15, 10, 5, 2, 1\}$.

7. Let $k \geq 3$ be an integer and consider the coin system $S = \{k+1, k, 1\}$. Show that on this system there is an amount for which the optimal solution uses 2 coins but the greedy algorithm will use k coins.

8. Suppose we have the following activities sorted according to their finishing time and with $C = [5; 8]$ meaning that activity C starts at time 5 and ends at time 8.

   - $A = [1; 4]$, $B = [3; 6]$, $C = [5; 8]$, $D = [6; 9]$, $E = [4; 9]$,
   - $F = [3, 10]$, $G = [9; 11]$, $H = [9; 12]$, $I = [12; 15]$.

   Your job is to find an optimal schedule for one room, that is, a schedule consisting of as many of the activities as possible and so that no two chosen activities overlap. Recall that this means that they cannot take place at the same time, but one may start just when the other finishes.

   - Find an optimal schedule using the greedy algorithm which always take the next activity as the one which has no overlap with the previously chosen ones and which ends as early as possible.
   - Try a different greedy strategy: always schedule that among the remaining activities which has no overlap with the already scheduled jobs and which has the largest starting time among such activities. Does it work? Can you prove that it always will work?

9. Apply the Huffman coding algorithm for finding an optimal prefix code for a text with the following characters and frequences (in % of the total number of characters).
   $a:$ 32; $b:$ 18; $c:$ 16; $d:$ 14; $e:$ 10; $f:$ 10

10. Discuss what is needed if we want to implement Huffman's algorithm.

11. Let K_5 be the edge weighted complete graph in Figure 2.

    - Construct a cycle on 5 vertices by starting in vertex number 1 and always including as the next vertex a non-chosen vertex that is as close as possible (measured by the cost of the edge between the two vertices) to the last vertex we included (so the next vertex after 1 will be 2 in the example). What is the cost of the cycle?
    - Is it an optimal solution? If not, produce one that is better.
    - Can you change the cost of just one edge so that the cycle we find by the greedy strategy can be arbitrarily bad compared to the cheapest cycle?

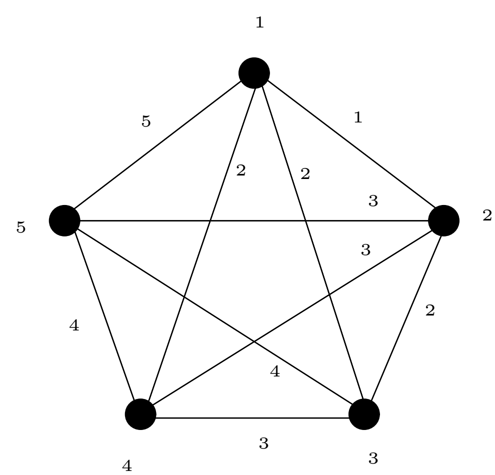

Figure 2: The graph K_5