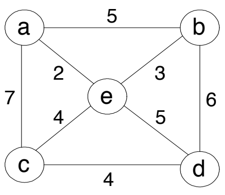
Werkcollege 12

You are allowed to answer in Dutch. Whenever an algorithm is required, it can be given in pseudocode or plain English (or Dutch), and its running time and correctness must always be justified (even informally, but in a clear way!).

12.1. Consider a long country road with houses scattered very sparsely along it. (We can picture the road as a long line segment, with an eastern endpoint and a western endpoint.) Further, let's suppose that the residents of all these houses are avid cell phone users. You want to place cell phone base stations at certain points along the road, so that every house is within four miles of one of the base stations.

Give an efficient algorithm that achieves this goal, using as few base stations as possible.

12.2. Run Prim's algorithm on the following graph



For each iteration, indicate: the vertex added to the spanning tree, the priority queue for the remaining vertices and, for each vertex, predecessor in the spanning tree and key.

12.3. Consider the problem of scheduling a set $S = \{a_1, a_2, \dots, a_n\}$ of activities. In the original problem, they had to be scheduled in one lecture hall. Now we allow for more lecture halls: we wish to schedule all the activities using as few lecture halls as possible. Give an efficient greedy algorithm to determine which activity should use which lecture hall.

12.4. Suppose you need to complete n distinct jobs, labeled $J_1, J_2, \dots, J_n$, which can performed completely independently of each other. Each job consists of two stages: first it needs to be *pre-processed* on a super-computer, and then it needs to be *finished* on a PC. Let's say that job J_i needs p_i seconds of time on the super-computer followed by f_i seconds of time on the PC.

Since there are at least n PC's available, the finishing of the jobs can be performed fully in parallel — all the jobs can be processed at the same time. However, the super-computer can only work on a single job at a time, so you need to work out an order in which to feed the jobs to the super-computer. As soon as the first job in order is done on the super-computer, it can be handed off to a PC for finishing; at that point in time a second job can be fed to the super-computer; when the second job is done on the super-computer, it can proceed to a PC regardless of whether or not the first job is done or not (since the PC's work in parallel); and so on.

Let's say that a *schedule* is an ordering of the jobs for the super-computer, and the *completion time* of the schedule is the earliest time at which all jobs will have finished processing on the PC's.

Give a polynomial-time algorithm that finds a schedule with as small a completion time as possible.

12.5. Suppose you are given a connected graph G, with edge costs that are all different. Prove that G has a unique minimum spanning tree.