Chair for Algorithms
and Data Structures
Prof. Dr. Hannah Bast

Efficient Route Planning
SS 2011

http://ad-wiki.informatik.uni-freiburg.de/teaching

Final Exam

Monday, August 15, 2011, 2.00 pm - 3.30 pm, HS 026 in building 101

General instructions:

There are four tasks, of which you can select three tasks of your choice. If you do all four tasks, we will only count the best three. Each task gives 10 points. You have one and a half hours of time overall, that is, 30 minutes per task on average. You need half of the total number of 30 points to pass the exam. The exact correspondence between the number of points and marks will be decided when we correct your exams.

You are allowed to use any amount of paper, books, etc. You are not allowed to use any computing devices or mobile phones, in particular nothing with which you can communicate with others or connect to the Internet.

On every page of your solutions, please write your Matrikelnummer and your name IN PRINTED LETTERS, and number the pages consecutively!

Good luck!

Task 1 (Dijkstra and A-Star, 10 points)

Consider the following graph (drawn in ASCII), with 22 nodes (drawn as `S` or `T` or `o`) and 24 undirected arcs (drawn as `|` or `---`). The cost of each arc is 1.

```
     o---o---o---o
     |           |
o    o       o---o
|    |       |   |
o----o   S---o   o
     |   |   |   |
     o---o   o   o
     |       |   |
o----o---o---o---T
```

1.1 Copy the graph on a sheet of paper and simulate a run of Dijkstra's algorithm from the node marked `S` (source) until the node marked `T` (target) is settled. It should be visible from your drawing: (1) which nodes were settled in which order (mark the settled nodes and write the order number nearby); (2) the tentative and final distances computed. (3 points)

1.2 Repeat task 1.1, but this time running A-Star, using the Manhattan-distance (defined in the next paragraph) as heuristic function. Instead of tentative distances, write the sum of the tentative distances and the heuristic function, making both summands explicit (for example, write $2 + 2$ and not 4). For this task you do not have to prove that A-Star finds an optimal task using this heuristic function, you can just assume it. (3 points)

The Manhattan distance is defined as follows. Think of the nodes above as drawn at the intersection points of a 5×5 *grid, with the upper left corner being intersection point* $(1, 1)$ *and the lower right corner being intersection point* $(5, 5)$*. Then the Manhattan distance between two nodes, at intersection points* (x_1, y_1) *and* (x_2, y_2)*, respectively, is simply* $|x_1 - x_2| + |y_1 - y_2|$*. For example, the Manhattan distance between the node marked* `S` *and the node marked* `T` *is* 4.

1.3 Prove that, in any grid graph of the kind above (edges only horizontally and vertically and between neighbouring grid points, all edge costs 1), the Manhattan-distance is a lower bound on the shortest path between two nodes. (4 points)

Task 2 (Contraction Hierarchies, 10 points)

2.1 Consider the same (grid) graph as in Task 1. Draw the graph after contracting all nodes of degree strictly less than 3. You can contract those nodes in an order of your choice, but please specify the order you chose (for that you can denote nodes by their grid coordinates, for example, the node marked `T` has coordinates 5, 5). Draw only the nodes remaining after these contractions and only edges between those nodes (original edges as well as shortcuts added due to the contractions). Don't forget to write the edge costs along with each arc. (3 points)

2.2 Consider the graph you have drawn for task 2.1. Contract all the remaining nodes such that as few shortcuts as possible are added. Clearly indicate the order in which you contract the nodes and which shortcuts were added when and why, or, when a node is contracted without adding a shortcut, why. (3 points)

2.3 What is the best and what is the worst order of contraction (with respect to the number of shortcuts added) for the cartwheel (auf Deutsch: Wagenrad) graph defined below? For both orders, specify the number of shortcuts that have to be added, and for each shortuct give a brief explanation why it has to be added, and for contracted nodes where no shortuct is added, why no shortcut needs to be added. (4 points)

The cartwheel graph has $n+1$ *nodes* $v_0, v_1, \ldots, v_n$ *and* $2n$ *edges, namely* $\{v_i, v_{i+1}\}$ *for* $i = 1, \ldots, n-1$*, and* $\{v_n, v_1\}$ *(the rim of the wheel, auf Deutsch: die Felge), and* $\{v_0, v_i\}$*, for* $i = 1, \ldots, n$ *(the crossing, auf Deutsch: die Speichen). The edges of the rim have cost* 2 *each and the edges of the crossing have cost* 1 *each.*

Task 3 (Multi-Label Dijkstra, 10 points)

3.1 Consider the following directed graph with bi-criteria costs. It has four nodes A, B, C, D. There is an arc from A to C with cost $(1, 1)$. There is an arc from A to D with cost $(5, 0)$. There is an arc from C to B with cost $(1, 1)$. There is an arc from D to B with cost $(1, 1)$. And there are arcs in both directions between C and D, both with cost $(2, 2)$.

Draw the graph and simulate an execution of multi-label Dijkstra from node A until all optimal labels for B are found. In your drawing, make sure to indicate the order in which the various labels were settled, and when a label is discarded and why. (3 points)

3.2 Prove that the elements of an arbitrary Pareto-set S of two-criteria costs, like above, can always be ordered such that the first components form a strictly increasing sequence and the seconds components form a strictly decreasing sequence. For example, for the Pareto-set $S = \{(1, 5), (2, 4), (7, 1), (4, 3)\}$ this (unique) order is $(1, 5)$, $(2, 4)$, $(4, 3)$, and $(7, 1)$. (3 points)

3.3 Write a function that, given an arbitrary Pareto-set S of two-criteria costs in the ordering described in 3.2, and an arbitrary additional cost c, computes the Pareto-set of $S \cup \{c\}$, again in the described ordering. Make use of the ordering so that your function makes fewer comparisons than would be necessary if the set was not ordered, and give a formula for the number of comparisons needed ($\Theta(...)$ suffices). (4 points)

For your function you can assume that you have a class Cost *which holds a two-criteria cost and has access methodes for both components, as well as a class* OrderedSetOfCosts *which holds an ordered set of costs, with methods to access an arbitry element and to add and delete elements.*

Task 4 (Transit Networks, 10 points)

Consider the following transit network.

- There are 6 stations A, B, C, D, E, F, and three buses 1, 2, and 3.
- Bus 1 takes the route A, B, D, E, starting at A at times 8:10, 8:40, 9:10, 9:40, ..., 20:10.
- Bus 2 takes the route A, C, D, F, starting at A at times 8:00, 8:30, 9:00, 9:30, ..., 20:00.
- Bus 3 takes the route D, E, F, starting at D at times 8:30, 9:30, 10:30, ..., 20:30.
- The time from one station to the next is 5 minutes for all bus for all trips.
- The time a bus spends at a station between arrival and departure at that station is zero.
- The transfer buffer is 5 minutes.

4.1 Draw the time-dependent graph for the whole transit network. For the arcs, you need not specify the cost functions, just label each arc with the bus lines which use it. (2 points)

4.2 Draw the following portion of the time-expanded graph for the transit network described above. The portion should contain the set of all nodes at station D between 8:00 and 8:30 (inclusive), all incoming and outgoing arcs of those nodes, and the nodes to which those arcs lead or where they come from. (3 points)

Draw the three types of nodes (arrival, departure, transfer) in different colors or using different symbols. Also draw the arcs adjacent to transfer nodes in different colors. Otherwise your picture will become a mess.

4.3 Using (travel time, number of transfers) as bi-criteria cost, determine the set of optimal transfer patterns for each of the following four station pairs: (A, B), (A, D), (A, E), (A, F). You can do the required shortest path computations in your head. As a proof it suffices that you write down, for each non-trivial transfer pattern, one optimal trip with that pattern. (3 points)

For example, a non-trivial optimal transfer pattern for the station pair (B, F) *is* B, D, F *and an optimal trip with that pattern is* $B@8{:}15 \rightarrow^1 D@8{:}20, D@8{:}30 \rightarrow^3 F@8{:}40$*, where* $\rightarrow^x$ *denotes a direct connection with bus* x.

4.4 Assume you are at station A at 9:10 (ready to board at that time) and you want to go to station F. Find all optimal connections for this query, using the bi-criteria cost function from task 4.3. If two connections have the same cost, you can take either or both. (2 points)