Solutions to exercise No. 6

Samuel von Baussnern

10. Mai 2013

1 Task B: Minimum Spanning Tree

1.1 Jarnik, Prim and Dijkstra algorithm

The algorithm starts with a single vertex of the graph. Then loops through all the edges of already marked vertexes and adds the edge, which satisfies two rules:

1. It has the minimum weight, and
2. wouldn't connect to an already added vertex.

This step is repeated until every vertex is added or no more edges satisfy the second rule.

1.2 Practical example

I number the vertexes of the graph clockwise (12 o'clock, 2 o'clock, 4 o'clock, etc.), for saving space I'll omit the 'o'clock'-term in the future.

The list of the added edges in chronological order starting the vertex 12:

1. (12, 2)
2. (2, 8)
3. (2, 6)
4. (12, 10)
5. (10, 4)

Results in this adjacent matrix:
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

The weight of the MST: 19. The weight of the graph: 51.

1.3 Time complexity

The time complexitiy depends heavily on the loop through the edges of the vertexes. Therefore it's crucial to use an appropriate data structure:

The algorithm is used in computer network design, in implementing efficient circuit design, clustering gen expressions, handwriting recognition of mathematical equations and of course as a pedagogical instrument to graphs and greedy algorithms, just to name a few.

Data structre used	Time complexity	Reason
Adjacency matrix	$\mathrm{O}\big(\lvert V\rvert^2$	Lookup is of constant time
Binary heap and adjacency list	$\mathrm{O}\big((\lvert V\rvert + \lvert E\rvert)\log\lvert V\rvert\big)$ $= \mathrm{O}\big(\lvert E\rvert\log\lvert V\rvert\big)$	Finding and deleting costs $\mathrm{O}\big(\log n\big)$
Fibonacci heap and adjacency list	$\mathrm{O}\big(\lvert E\rvert + \lvert V\rvert\log\lvert V\rvert\big)$	Finding is of constant cost and deleting the min has a amortized cost of $\mathrm{O}\big(\log n\big)$

2 Task C: Kruskal

2.1 Use of Kruskal's algorithm

Since the weights are pair-wise disjoint there exists only one MST.
Listing the chosen edges:

1. (a,f)
2. (c,d)
3. (c,e)
4. (f,d)
5. (b,c)

2.2 Another example

Six different MSTs:
Listing the chosen edges:

1. (a,b)
2. (a,c)
3. Choice between:
   (a) (a,d)
   (b) (c,d)
4. Choice between:
   (a) (e,d)
   (b) (e,b)