1 Exercises

1.1 Spanning trees

1. Consider a set of 5 towns. The cost of constraction of a road between towns i and j is a_{ij}. Find the minimum cost road network connecting the towns with each other.

$$\begin{bmatrix} 0 & 3 & 5 & 11 & 9 \\ 3 & 0 & 3 & 9 & 8 \\ 5 & 3 & 0 & +\infty & 10 \\ 11 & 9 & +\infty & 0 & 7 \\ 9 & 8 & 10 & 7 & 0 \end{bmatrix}$$

2. Prove the following propositions.

   (a) Let e be a minimum weight edge in a graph G. Then (u, v) is contained in a minimum weight spanning tree of graph G.

   (b) Let e be a maximum weight edge (of the cycle) in a cycle of a graph $G = (V, E)$. There exists a minimum weight spanning tree of graph $G' = (V, E \setminus \{e\})$ which is also a minimum weight spanning tree of graph G.

   (c) Let T be a minimum weight spanning tree of graph $G = (V, E)$, and let V' be a subset of V. Let T' be a sub-graph of T induced by V', and let G be a sub-graph of G' induced by V'. If T' is connected then T' is a minimum weight spanning tree of graph G'.

   (d) Given a graph $G = (V, E)$, let V' is a strict subset of V. Let e be a minimum weight edge which connects V and $V \setminus V'$. There exists a minimum weight spanning tree which contains e.

3. Let e be an arbitrary edge in a graph G. Is it always possible to construct a spanning tree of G which contains e? And if we are given a set of edges?

4. In some problems we want that certain pairs of vertices are directly connected with each other. Modify the Prim's algorithm in order to solve the minimum spanning tree problem with this additional constraint.

1.2 Shortest paths

1. A contractor assigns to one of his building sites a changing number of qualified workers between March and August:

Month	March	April	May	June	July	August
Stuff	4	6	7	4	6	2

Workers can be discharged from the building site only in the beginning of the month. Suppose that in February and in September there is exactly three workers at the building site.

The aim of the contractor is to plan an allocation of workers which minimizes the sum of the following costs :

- Transfer cost. Hiring a worker to the building site costs 50 euros and discharging a worker costs 80 euros.
- Transfer rules. The contractor can hire at most 3 workers at a time and can discharge at most one third of his stuff at a time.
- Over and underpopulation costs. A superfluous worker costs 100 euros, whereas missing of one worker costs 200 euros. When some stuff is missing, workers take additional working hours, but they do not accept to work additionaly more than 1/4 of their normal time.

Formulate this problem as a shortest path problem and solve it using the Dijsktra's algorithm.

2. You possess a bank-note of p euros and you want to change it to coins of $a_1, a_2, \ldots, a_n$ euros. Is it possible? If yes, what is the minimum number of coins? Formulate this problem as a shortest path problem and solve it using the Dijsktra's algorithm.

3. Let $d_k(j)$ be the length of the shortest path between from vertice s to k with k edges at most. Find a recursion for $d_k(j)$. Prove that $d_{n-1}(j) = D(j) =$ the shortest distance from s to j. Here n is the number of vertices in the graph.

2 Solutions

2.1 Spanning trees

1. To find the minimum cost road network, we formulate the problem as a minimum weight spanning tree problem. Consider the complete graph with 5 vertices. The weight of edge (i, j), $i, j \in \{1, \ldots, 5\}$, is set to a_{ij}. Now we find the minimum weight spanning tree of the graph constructed. The tree found is $\{(1, 2), (2, 3), (2, 5), (4, 5)\}$ and the total cost is 21.

2. (a) Let T be a minimum weight spanning tree in graph G and T does not contain edge $e = (u, v)$. We add edge e to the spanning tree T. By the property of trees, T now contains a cycle and e is one of edges in this cycle. Now we remove from T an arbitrary edge $e' \neq e$ which belongs to the cycle. We obtain a new spanning tree T'. The weight of spanning tree T' is not more than the weight of spanning tree T, as the weight of e is not more than the weight of e'. Therefore T' is also a minimum weight spanning tree in graph G and T' contains e.

(b) Let T is a minimum weight spanning tree in graph G and T contains edge e. T cannot contain all the edges from the cycle and we can replace in T the edge e by another edge e' which belongs to the cycle and is not contained in T. We obtain a new spanning tree T'. The weight of the spanning tree T' is not more than the weight of the spanning tree T, as the weight of e is not more than the weight of e'. Therefore T' is also a minimum weight spanning tree in the graph G. Moreover T' is also a minimum weight spanning tree in the graph $G' = (V, E \setminus \{e\})$, as T' does not contain the edge e and graphs G and G' have the same number of vertices.

(c) Suppose that T' is not a minimum weight spanning tree in graph G' and S' is a minimum weight spanning tree in G'. Then, if we joined the subset of edges $T \setminus T'$ to S', then we would obtain a spanning tree S in the graph G. The weight of S would be smaller than the weight of T and this contradicts the condition that T is a minimum weight spanning tree. Thus, our assumption is false and T' is a minimum weight spanning tree in the graph G'.

(d) Let T is a minimum weight spanning tree in a graph G and T does not contain the edge e. We add e to the spanning tree T and obtain a cycle. This cycle should contain besides e another edge e' which connects subsets of vertexes V' and $V \setminus V'$. We remove e' from T and obtain a spanning tree T'. The weight of T' is not more than the weight of the spanning tree T, as the weight of e is not more than the weight of e'. Therefore T' is also a minimum weight spanning tree in the graph G and T' contains e.

3. It is easy to see that it is always possible if and only if G is connected. If G is not connected then there is no connected sub-graph of G, therefore there is no spanning tree. Suppose now that G is connected. Consider a spanning tree T of G. If e is in T, we are done. Otherwise we add e to T. By the property of trees, T now contains a cycle and e is one of edges in this cycle. Now we remove from T an arbitrary edge $e' \neq e$ which belongs to the cycle. We obtain a new spanning tree T' which containts e.

   If we are given a set S of edges then it is possible to construct a spanning tree containing this set if and only if G is connected and no subset of S is a cycle. To construct a spanning tree containing set S of edges, we again start from an arbitrary spanning tree T of G and proceed iteratively. On each iteration, we insert an edge from $S \setminus T$ and then delete an edge in $T \setminus S$ from the cycle formed in T.

4. There are several ways to modify the Prim's algorithm in order to take into account this additional constraint. One of them is the following. Given a graph $G = (V, E)$, let S be the set of edges that represents our additional constraint, i.e. S is the set of edges which should be included in the spanning tree. Now set the weight of all the edges in S to some value which is less than the minimum weigth of edges in $E \setminus S$. Then we

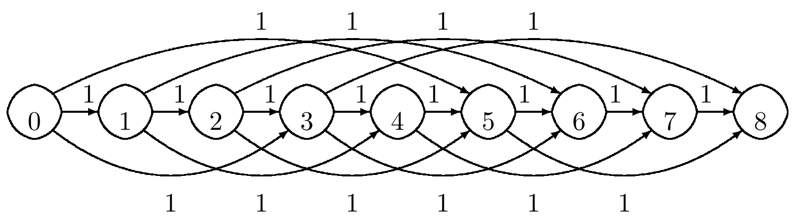
run the Prim's algorithm on the graph with modified weights. The cost of the spanning tree found is calculated using the original weights.

2.2 Shortest paths

1.

2. Consider the graph G which has $p+1$ vertices: $0, 1, 2, \ldots, p$. For each coin a_i there are next edges in G: $(0, a_i)$, $(1, a_i + 1)$, $(2, a_2)$, …, $(p - a_i, p)$. The length of each edge is 1. Now, the length of the shortest path in the obtained graph G gives us the answer to our initial problem. If there is no path from vertex 0 to vertex p, then it is impossible to change the bank-note.

   For example, if we have: $p = 8$, $a_1 = 1$, $a_2 = 3$, $a_3 = 5$, then we have the next graph G:



   The shortest path in the graph is $0 \to 5 \to 8$ or $0 \to 3 \to 8$ and length is 2. So, if we have pieces of 1, 3 and 5 euros and want to change 8 euros, minimum 2 pieces should be used.

3. The recursion is:

$$d_k(j) = \min \left\{ d_{k-1}(j), \min_{i:\ (i,j) \in E} \{ d_{k-1}(i) + c_{ij} \} \right\}.$$

   The proof of the fact that $d_{n-1}(j)$ is the length of the shortest path between s and j follows from the observation that a shortest path between s and j has at most $n - 1$ edges.