Advanced Algorithms - Homework 2

Question 1

First, we need to find the biggest size of vertices forming a maximal independent set (MIS), note this size by **k**.

In order to find k we can perform a binary search in the range [1,n] using the function that runs in polynomial time.

**Note** - The first TRUE that returns is not enough, we need to keep searching for the biggest.

**Runtime** to find the best k:

After we know the value of k we can start the second part of the solution, run iterations and each iteration we'll delete a vertex and check with if there is still MIS by the size of k and if the graph has k left vertices we can quit the loop and return the graph, if there is not MIS by the size of k (the decision function returns FALSE) then we need to undo the deletion of the vertex and continue to the next iteration.

Pseudo:

\* check edge cases

k = binary\_search(G) # as described above

for v in V:

G.delete\_vertex(v)

if :

if :

return vertices & edges

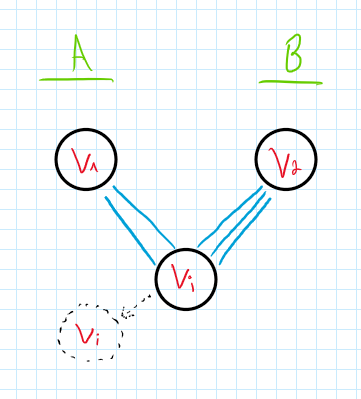
else:

G.add\_vertex(v)

**Runtime** of :

Question 2

First, we need to understand how the algorithm works, lets see the example below



As we can see and

The algorithm works in a greedy way 🡪 in this example the algorithm will decide to

assign to group A. We can see that the worst scenario is if , therefore half of the edges will be in the maximum cut and half will be "lost" ()

**Note** - the worst case for this algorithm is when the equality I just mentioned above repeats itself for each of the vertices and their edges equal distributed to (A, B respectively) without edge between and

|  |  |
| --- | --- |
| Example for the worst case | The algorithm will produce maximum cut size of **2** but the optimal solution is **4** |
|  |  |

We can see the simplest example above.

Question 3

Let be a **complete** graph with weights that obey the **triangle inequality**, and a subset of vertices (terminals).

Let be the cost of an optimal solution to Steiner tree problem.

We'll start by doubling each edge to obtain an Euler cycle, with DFS tour we know this cost is .

Making a Hamilton circuit using "short-cutting" Steiner vertices and visited vertices (shortcut = connecting new edge between pair of adjacent terminals in the DFS).

The short-cutting doesn’t increase the total of the cost (complete graph & triangle inequality). Deleting one edge from this Hamilton circuit yields a spanning tree of with cost at most .

So, if this tree is a MST on we know its cost is less than

**To summary:**

= cost of the tour on the terminals in

cost of any spanning tree (the tree after the manipulations, adding edges short-cutting and so on)

cost of the MST of the tree

Question 4

Let be the number of the bins after using the algo Next-fit and be the number of bins possible that needed to pack all of the items.

Two things we know:

1. The maximum size of each item is
2. Each of the bins full at least of the maximum capacity (max cap. = 1) (from 1.)

The total size of the items is at least 🡪 Order the equation will results us that

Let be a sequence as the following where .

Consider our instance as therefore we have of the items and we have of the items .

The algo Next-fit will pack this instance using bins and the will do with

bins.

exactly as Next-fit algo.

**Note** - derive from the last bin (in optimal solution),

Question 5

I read and understood the example and the tightness of the analysis.