Compulsory Exercise, Algorithms Number 2

Jonas Feierabend & Sebastian Nausester

**Problem 1**

Ein Bild, das Text, Screenshot, Reihe, Schrift enthält.

Automatisch generierte Beschreibung

1. Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

   Automatisch generierte BeschreibungLocal search

* No further improvements possible
* Not optimized it takes 42 time units (adding all jobs )
* Optimized it takes 15 time units (the time m2 needs)

1. Greedy

Ein Bild, das Text, Screenshot, Zahl, Reihe enthält.

Automatisch generierte Beschreibung

* Finished after 16 time units

1. Longest processing time

Ein Bild, das Text, Screenshot, Zahl, parallel enthält.

Automatisch generierte Beschreibung

* Finished after 14 time units (best)

1. The lower bound is (The Total time of the jobs) / (number of machines). You cant do it faster because you need to calculate every job and the fastest way would be 100% parallel -> then you can just devide by the number of machines
2. It can be possible. For example if you have 3 jobs with 6 time units and 3 machines. The optimal solution is trivial

But there are also instances where it is not possible. For example job 1 with 10 time units and job 2 with 5 time units. With 2 machines. As long as you cant break apart one job there is no better solution than start to calculate job 1 immeadiately -> you need 10 time units (and not (10+5)/2 = 7.5 )

1. Ein Bild, das Text, Screenshot, Schrift enthält.

   Automatisch generierte Beschreibung

Ein Bild, das Reihe, Diagramm, Kreis enthält.

Automatisch generierte Beschreibung

* 1. Triangle inequality for vertices 1 to vertices 5

1. Nearest addition
2. Build a tour between the closest cities
3. Find closest vertices

S={3,6,1} Add it after 3 in the Tour

1. Find closest vertices

S = {3,6,1,2} Add 2 after 1 and before 6

1. Double Tree
2. Find MST with Kruskal

|  |  |
| --- | --- |
| Edge | Weight |
| (1,2) | 4 |
| (1,3) | 5 |
| (1,4) | 10 |
| (1,5) | 11 |
| (1,6) | 7 |
| (2,3) | 9 |
| (2,4) | 6 |
| (2,5) | 7 |
| (2,6) | 9 |
| (3,4) | 9 |
| (3,5) | 6 |
| (3,6) | 2 |
| (4,5) | 3 |
| (4,6) | 9 |
| (5,6) | 6 |

|  |  |
| --- | --- |
| Edge | Weight |
| (3,6) | 2 |
| (4,5) | 3 |
| (1,2) | 4 |
| (1,3) | 5 |
| (2,4) | 6 |
| (3,5) | 6 |
| (5,6) | 6 |
| (1,6) | 7 |
| (2,5) | 7 |
| (2,3) | 9 |
| (2,6) | 9 |
| (3,4) | 9 |
| (4,6) | 9 |
| (1,4) | 10 |
| (1,5) | 11 |

While (add smallest edge as long as there are no circles):

S+= edge

1. S = {(3,6)}
2. S = {(3,6),(4,5)}
3. S = {(3,6),(4,5),(1,2)}
4. S = {(3,6),(4,5),(1,2),(1,3)}
5. S = {(3,6),(4,5),(1,2),(1,3),(2,4)}
6. S = {(3,6),(4,5),(1,2),(1,3),(2,4)} don’t add (3,5) , (5,6) , (1,6) , (2,5), …
7. Double all edges

Ein Bild, das Entwurf, Diagramm, Zeichnung, Lineart enthält.

Automatisch generierte Beschreibung

1. Find a euler tour
2. Skip nodes which you already visited
3. Christopids algorithm
4. Find MST (from c) S = {(3,6),(4,5),(1,2),(1,3),(2,4)}
5. Find perfect matching for odd degree vertices (6 & 5)

* (5,6)

1. Find Euler Tour