Gruppe : Nikolic , Nofal, Hattinger

Aufgabe 34

1.Kruskals Algorithmus kann wie folgt optimiert werden:

Die asymptotische Laufzeitkomplexität des MST-Algorithmus nach Kruskal hängt unter anderem von der asymptotischen Laufzeitkomplexität des Sortier-algorithmus für die aufsteigende Sortierung der Kantengewichte ab. Hier wurde in der Vorlesung die allgemeine Problemkomplexität für das Sortieren mit $O(n \cdot \log(n))$ angegeben. Durch die Einschränkung laut Angabe besteht in diesem Punkt allerdings das Optimierungspotenzial. Durch die Einschränkung der Kantengewichte auf das Intervall $[1, |V|]$ kann hier Counting Sort verwen-det werden, was die Laufzeitkomplexität des Sortierens auf $\Theta(|E| + |V|)$ re-duziert. Somit ergibt sich eine Gesamtlaufzeit von $O(|V| + |E| + |V| \cdot \log(|V|)) = O(|E| + |V| \cdot \log(|V|))$.

Ebenso kann Counting Sort bei jedem beliebigen, endlichen Intervall $[1, W]$ verwendet werden, wobei hier eine asymptotische Laufzeit von $O(W + |E|)$ besteht, womit sich wiederum eine gesamte asymptotische Laufzeitkomplexität von $O(W + |E| + |V| \cdot \log(|V|))$ ergibt.

aufgabe 35

```
Floyd-Warschall(W,n)
for K ← 1 to n do
    for i ← 1 to n do
        for J ← 1 to n do
            d_{i,J}^{(K)} ← min(d_{i,J}^{(K-1)}, d_{i,K}^{(K-1)} + d_{K,J}^{(K-1)})
for i ← 1 to n do
    if d[i,i]<0 then
        return("Graph contains Negtive cycles")
```

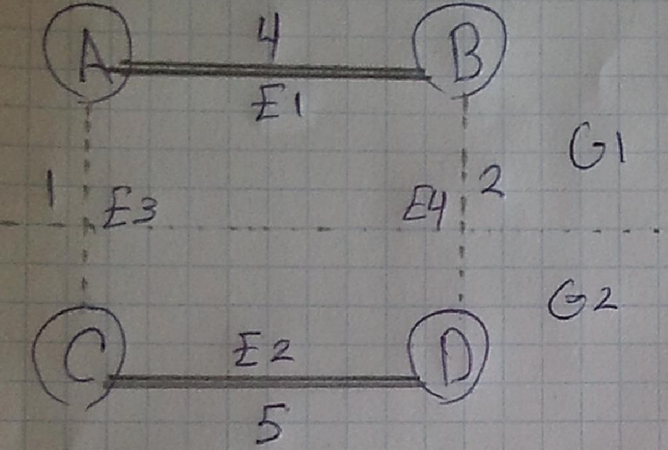
- die idee dahinter wenn der algorithmus fertig ist wird die diagonale von dem entstehnden matrix auf negative werte überprüft, wird einen wert ~~mit~~ gefunden, so wird der obengenannter string zurückgegeben "Graph contains Negative cycles".

Aufgabe 3 b

der devide & conquer Algorithus ist nicht korrekt

bzw liefert keinen mindimalen spannbaum

gegen Beispiel



angenommen wir haben den folgenden graphen $G = (V, E)$ der aus 4 Knoten besteht, also $V = \{A, B, C, D\}$

$E = \{(A,B), (B,D), (D,C), (C,A)\}$

wird in 2 mengen Partioniert, $G_1 = \{(A,B)\}$ und die menge $G_2 = \{(C,D)\}$, wobei die Kante "E1" die, die 2 Knoten A, B in G_1 gewicht = 4 hat, und die Kante "E2" die, die 2 Knoten C, D in G2 gewicht = 5 hat, nun laut der Algoithmus die Kante E3 wird mit der Kante E4 verglichen, da E3 ein kleineres gewicht als E4 hat, verbindet die Kante E3, die Kanten E1 mit E2

$\Rightarrow$ $E1 + E3 + E2 = 4 + 1 + 5 = 10$ als mindimals gewicht und E1, E2, E3 als minimal spannbaum, das ist aber falsch, weil wie zu sehen ist der minimal spannbaum ist $E1, E3, E4 = 4 + 2 + 1 = 7$ als minal gewicht

$\Rightarrow$ der Algorithmus lieft nich den minimalen spannbaum