Lulu Eve
260473661

See order please! thanks

Assignment 2

① algorithm
② complexity
③ correctness
④ Greedy choice
⑤ Substructure prop.

EX 3: Greedy algorithm

$\{x_1, x_2, \ldots, x_n\}$

① Our algorithm will be several steps.

i) Sort the list of points such that $x_1 \leq \ldots \leq x_n$

sort (Set)

ii) while (set) != empty {

i) Take the leftmost point in the list
ii) Examine the interval $[x_1, x_1+1]$
iii) Remove all points $x_1, x_{(1+1)} \ldots x_{(1+k)}$ such that $x_{1+k} - x_1 \leq 1$ is taken from the list.
iiii) repeat on all points, return S'

(see step 4)

② Upperbound of algorithm: the above algorithm is an $O(n)$ algorithm upperbound.

③ Proof of Correctness:

We will proove this proof by looking at the contrary. Lets look at step ii i, or when we look at the leftmost point x_1. We could look at the interval $[x_1 - \alpha, x_1 + \beta]$ such that

- $\alpha \geq 0$
- $\beta \geq 0$
- $\beta - \alpha = 1$

} conditions

Since x_1 is the leftmost point → in current

Luau Eve
260473661

b/a

Set of points. From this junction, we know we will not cover any points in the interval $[x_1-\alpha, x$
Therefore, we will maximize the total number of possible points covered only if ~~and~~:
we select the interval

$$-[x_1-\alpha, x_1+B]$$
such that $\alpha=0$, $B=1$

(4) The Greedy Choice: suppose: S = our optimal solution
Suppose:
S places its leftmost interval at $[x, x+1]$
$\Rightarrow$ greedy choice is $x \leq x_1$ since it puts it first point as far right while still cove x_1

let S' = scheduled obtained by starting a replacing $-[x, x+1]$ by $[x_1, x_1+1]$

Argue: all points in S range $[x, x+1]$ are now covered by $[x_1, x_1+1]$ (the

The S' range does not cover $[x, x+1$
but $[x_1, x_1+1]$ = left most positio (doesn't effect S' as optimal solut

Conclusion: S' is the greedy choice (has same number of points as S)

Lucia Eve
260473661

⑤ Substructure Prop.

Let P = original problem
P has S as optimal greedy choice

Include $[x_1, x_1+1]$ (interval)
$P' \Rightarrow$ find points to right of x_1+1
S' = optimal solution for P'

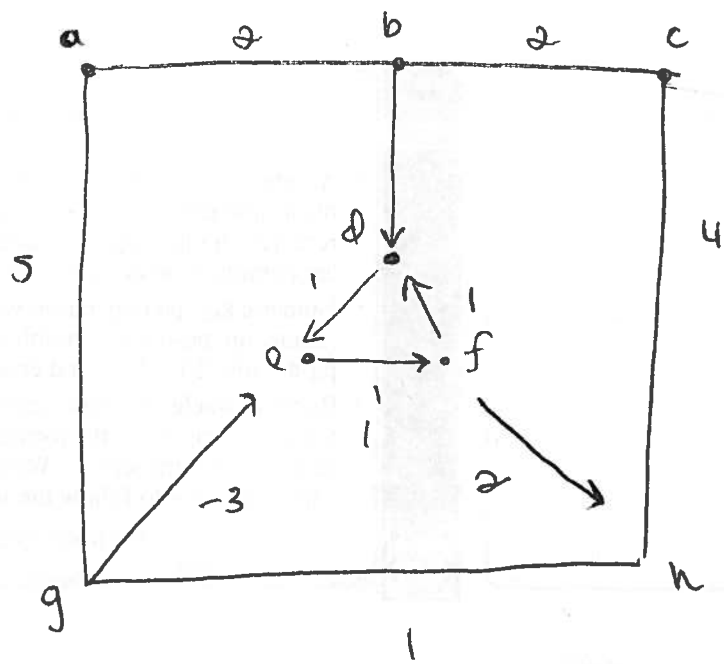
Since $cost(S) = cost(S') + 1$

Conclusively $\rightarrow$ an optimal solution P includes an optimal solution to P'

260473661

Exercise 4:

An example of a directed graph that produces the incorrect answers is below.



Since edge (g,e) is of negative weight -3, the path of the combined (a,g,e) is 5-3=2. It is better than the weight of the finalized short path (a,b,d,e) which is of total weight 4. Edge (e,f) has been relaxed. There is no way for Dijska algorithm to restanalized.

Because of this, it is an incorrect implementation.

Lucia Eve
260473661

Ex. 5 Bipartite Graphs

$G = (V, E)$ is bipartite if and only if
G has no odd cycle

ONLY IF: Suppose:
$G = (L, R, E) \leftrightarrow$ bipartite
let $v_0, \ldots, v_{k-1}, v_k = v_0 \leftrightarrow$ cycle in G

Then suppose:
$v_0 \in L$ then
$v_1 \in R$ since $\{v_0, v_1\} \in E$
then
$v_2 \in L$ since $\{v_1, v_2\} \in L$

if we continue, we can see that
if $i == odd$
then $v_i \in R$ ~~and~~

but
if $i != odd$
then $v_i \in L$

Thus, since $v_k = v_0 \in L \rightarrow$ implies that k is even. The cycle is of even length.

IF Suppose:
G has no cycles of odd length

Assume:
G is connected

Pick a vertex $u_0 \in V$

For every vertex $v \in V$
let p_v be the path $u_0 \to v$
also let d_v be the length of this pa[th]

Set:

$L = \{ v \in V \mid d_v \text{ is even} \}$
AND
$R = \{ v \in V \mid d_v \text{ is odd} \}$

$\downarrow$

$V = L \cup R$ is a partition of V

Show:
(L, R, E) is bipartite

if not, then there is some $\{u, v\} \in E$ suc[h]
that both $u, v \in L$ or both $u, v \in R$.
In either case, there is a closed path in G gi[ven]
by $p_u, \{u, v\}, p_v$ (that is from $u_0 \to u$, th[en]
$u \to v$, then $v \to u_0$) whose total length $= l$
is odd.
Since G has a closed walk of odd
then G also has a cycle of odd length
This is contradictory.
Thus, $G = (L, R, E)$ is bip[artite]