Homework 1

Mitar Milutinovic (24090156)

February 19, 2013

1.

1.(a)

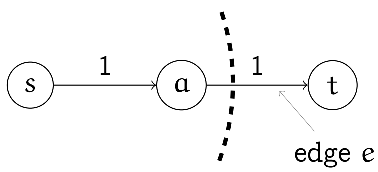
$$T_1(n) = O(n^2)$$

$$T_2(n) = O(n^3)$$

1.(b)

$$\begin{aligned} T(n) &= 32T\left(\frac{n}{4}\right) + 67O\left(\left(\frac{n}{4}\right)^2\right) = \\ &= 32T\left(\frac{n}{4}\right) + O\left(n^2\right) = \\ &= O\left(n^{\log_4 32}\right) = O\left(n^{2.5}\right) \end{aligned}$$

1.(c)



Max-flow solution for the graph above has a minimum cut which contains an edge e with capacity 1. But even if we increase capacity of the edge e, max flow does not increase. Thus, the problem statement does not hold by counterexample.

1.(d)

$$
\begin{aligned}
&\max z\\
-x-y\qquad\quad &+z\leqslant 0\\
-y-w&+z\leqslant 0\\
-3x\qquad -w&+z\leqslant 0\\
x+y+w\qquad &=1\\
x,y,w&\geqslant 0
\end{aligned}
$$

1.(e)

We relax $\forall v\in V, x_v\in\{0,1\}$ into $\forall v\in V, x_v\geqslant 0$. Solving the linear program gives us $\tilde{x}_v$. Vertex cover of G is $S=\left\{v|\tilde{x}_v\geqslant\frac{1}{2}\right\}$. S is a 2-approximation for the optimal vertex cover of G.

1.(f)

$$A=C\left(f(s_I),t\right)+v_I$$

$$B=C\left(t,f(s_I)\right)+v_I$$

1.(g)

$O(nE)$, where $E=\max_i e_i$, the finishing time of the last job.

2.

2.(a)

function NOTDOMINATE(A)
    if $|A|=1$ **then**
        return A
    else
        $B\leftarrow$ NOTDOMINATE($A[0..(\frac{|A|}{2}-1)]$)
        $C\leftarrow$ NOTDOMINATE($A[\frac{|A|}{2}..(|A|-1)]$)
        $R\leftarrow[]$
        for b in B **do**
            if b is not dominated by any p in C **then**
                $R\leftarrow R+[b]$
        $R\leftarrow R+C$

return R

$A \leftarrow$ sort S by x coordinate in the ascending order
return NOTDOMINATE(A)

2.(b)

$A \leftarrow$ sort S by x coordinate in the decreasing order
find the longest increasing subsequence of y coordinates of points in A
return points of the longest increasing subsequence

3.

Each clause C_j contains three variables, x_i or $\bar{x}_i$. Each clause vertex c_j corresponds to clause C_j. Each variable vertex v_i corresponds to variable x_i, v_i' to variable $\bar{x}_i$. We connect every c_j to all v_i and v_i' except v_i if $x_i \in C_j$ and v_i' if $\bar{x}_i \in C_j$. We color the graph with $k = n + 1$ colors.

4.

4.(a)

$r \leftarrow \{\}$
for $i = 0 \to (n - (m-1) - 1)$ **do**
    $z \leftarrow 0$
    for $j = 0 \to m - 1$ **do**
        if $s_2[i + j] \neq s_1[j]$ **then**
            $z \leftarrow z + 1$
    if $z \leqslant k$ **then**
        $r \leftarrow r \cup \{i\}$
return r

4.(b)

$r \leftarrow \{\}$
$s_1 \leftarrow (s_1 - 0.5) \cdot 2$
$s_2 \leftarrow (s_2 - 0.5) \cdot 2$
$S_1 \leftarrow \text{FFT}(s_1)$
$S_2 \leftarrow \text{FFT}(s_2)$
$C \leftarrow S_1 \cdot$ REVERSE(S_2) $\triangleright$ correlation

```
c ← FFT(C)                                  ▷ c is of length n + m − 1
for i = m − 1 → n − 1 do
    if c[i] ⩾ m − 2 · k then
        r ← r ∪ {i − (m − 1)}
return r
```

5.

5.(a)

$$b(\pi)/2$$

5.(b)

```
while b(π) > 0 do
    for all π_i ... π_j between two breakpoints do
        if ρ(i, j) removes more breakpoints of π than ρ then
            ρ ← ρ(i, j)
        else if ρ(i, j) removes equal number breakpoints of π as ρ then
            if π_i ... π_j is increasing then
                ρ ← ρ(i, j)
    π ← π ∘ ρ
return π
```