AALG Self Study 1

Andreas Petersen
Mads E. Kalør
Søren B. Ranneries
Room 1.1.34

11. februar 2015

1 Exercise 1

A is an adjacency matrix of the graph. s_a.size is the size of the resulting list of the topological sort. .out on some vertex is the outgoing edges of that vertex. .to of some edge is the vertex that the edge leads to. $.w$ on some edge is the weight of that edge.

The recurrence for the algorithm is shown in Equation 1.

$$c(A,s,t) = \begin{cases} 0 & \text{if } s = t \\ -\infty & \text{if } s.out < 1 \text{ and } s \neq t \\ \max\limits_{e \in s.\text{out}} \{e.w + c(A, e.\text{to}, t)\} & \text{otherwise} \end{cases} \tag{1}$$

```
LongestPath($A, s, t$)
    s_a ← topological_sort($A$)
    for $i$ ← s_a.size to 1
        if s_a[$i$] = $t$ then
            $c$[s_a[$i$]] ← 0
        else if s_a[$i$].out.$w$ < 1 then
            $c$[s_a[$i$]] ← $-\infty$
        else
            max_val ← $-\infty$
            for $e \in$ s_a[$i$].out
                if $e.w + c[e.\text{to}]$ > max_val
                    $c$[s_a[$i$]] ← $e.w + c[e.\text{to}]$
                    $e$.to.par ← s_a[$i$]
    return $c[t]$
```

Topological sort is $\Theta(|V| + |E|)$. The total running time of LongestPath is $\Theta(|V|^2)$.

2 Exercise 2

The recurrence can be seen in Equation 2.

$$c(s,i,j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i > j \\ 2 + c(s, i+1, j-1) & \text{if } s[i] = s[j] \\ \max(c(s,i+1,j),\ c(s,i,j-1)) & \text{otherwise} \end{cases} \tag{2}$$

```
LongestPalindrome($s, i, j$)
    $c \leftarrow [1..n, 1..n]$
    $p \leftarrow$ new list
    for $j \leftarrow 1$ to len($s$)
        for $i \leftarrow j$ to 1
            if $i = j$ then
                $c[i,j] \leftarrow 1$
            else if $s[i] = s[j]$ then
                if $j - i = 1$
                    $c[i,j] \leftarrow 2$ # $c[i+1, j-1] = 0$ in this case
                else
                    $c[i,j] \leftarrow 2 + c[i+1, j-1]$
                    $p.add(i)$
                    $p.add(j)$
            else
                $c[i,j] = \max(c[i+1,j],\ c[i,j-1])$
    return $\langle c[1,n],\ p\rangle$
```

The complexity of LongestPalindrome is $\frac{1}{2}|s|(|s|+1) = \Theta(|s|^2)$.

3 Exercise 3: Exam Sheet Exercise 1

3.1 Sub Exercise 1

If the emp cannon is activated during the 2nd second, it will kill $\min(10, f(2-0)) = 3$ robots. It does not matter if the cannon is fired during the third or the fourth second. If the cannon is activated during the third second, it will only have charged up enough to kill 1 robot. If it is fired during the fourth second only, it can kill 3 robots, but there is only 1 robot to kill. The optimal strategy is to activate it during the third and fourth second. It will then kill a total of 5 robots.

If the cannon is not activated during the 2nd second it will kill no robots during the second second. If the cannon is fired during the third second it will kill $\min(10, f(3-0)) = 6$ robots. It can then be fired again the fourth second and kill 1 robot. It does not pay of to save the cannon for fourth second, as there is only one robot to kill then. The optimal strategy is then to fire the cannon at the third and fourth second, killing a total of 7 robots.

If the cannon is activated during the first second it will kill $\min(1, f(1-0)) = 1$ robot. Then the optimal strategy is to fire the cannon during the third second, where it will kill $\min(10, f(3-1)) = 3$ robots. It can then be fired again in the fourth second to kill 1 robot, resulting a total of 5 kills.

3.2 Sub Exercise 2

On the input shown in Table 1 the Schedule-Cannon will give the incorrect answer. It will fire during the 1, 4 and 5 second killing 7 robots. The optimal solution is to fire during the 3, 4, and 5 second killing a total of 8 robots.

i	**1**	**2**	**3**	**4**	**5**
x_i	1	5	10	5	1
$f(i)$	1	3	6	9	12

Tabel 1: Input to Schedule-Cannon