Assignment 11

I

There are three options:

- not add the i^{th} store:

$$profit[i] = profit[i-1]$$

- add the i^{th} store:

$$profit[i] = (profit[i-k] + profit[i]) \text{ max over } k$$

with $d[i] - d[k] > min$

- or just $profit[i]$

```
Max_profit(array m, array p, int min) {
    profit ← new array;
    profit[0] ← p[0];
    for (int i = 1; i < p.length; i++) {
        profit[i] = max(p[i], p[i-1])
        int j = 0;
        while (m[i] - m[j] >= min) {
            if (p[i] + profit[j] > profit[i]) {
                profit[i] = p[i] + profit[j];
            }
            j++;
        }
    }
    return profit[n];
}
```

II

a.

$$P[i,j,k] = \begin{cases} 1 \text{ if } i=0,\ j=0 \text{ and } k = M[i,j] \\ P(i-1,j,k-M[i,j]) + P(i,j-1,k-M[i,j]) \end{cases}$$

b $P[m+1, n+1, k] \longleftarrow$ new array (initialised to 0)

```
for (a=1 to m) {
    for (b=1 to n) {
        if (a == 1 == b) {
            P[a, b, m[a-1, b-1]] ++
        }
        else {
            for (c=0 to k) {
                PP[a, b, c] = P[a, b-1, c-M[a,b]] +
                              P[a-1, b, c-M[a,b]];
            }
        }
    }
}
return M[m+1, n+1, k];
```

III

a.

$$L[i,j] = \begin{cases} L[i+1, j-1] + 2 & \text{if } L[i] == L[j] \\ L[i+1, j] & \text{if } L[i] \neq L[j] \text{ and } L[i+1, j] \geq L[i, j-1] \\ L[i, j-1] & \text{else} \end{cases}$$

b.

```
compute_L (array s) {
    for (i=1 to n) {          (n = length s)
        c[i,i] = 1;
    }
    for (j=1 to n-1) {
        for (i=1 to n-j) {
            if (s[i] == s[i+j]) {
                c[i, i+j] = 2 + c[i+1, i+j-1]
            }
            else {
                c[i, i+j] = max( c[i+1, i+j],
                                 c[i, i+j-1])
            }
        }
    }
}
```