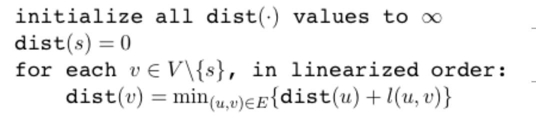
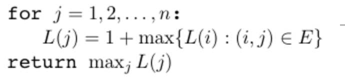
Chapter 6. Dynamic Programming

6.1 Shortest Paths in dags, revisited



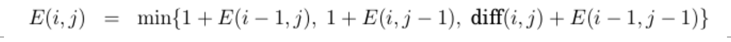
```
initialize all dist(·) values to $\infty$
dist($s$) $= 0$
for each $v \in V \backslash \{s\}$, in linearized order:
    dist($v$) $= \min_{(u,v) \in E}\{$dist($u$) $+ l(u,v)\}$
```

6.2 Longest increasing subsequences



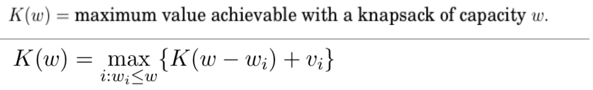
```
for $j = 1, 2, \ldots, n$:
    $L(j) = 1 + \max\{L(i) : (i,j) \in E\}$
return $\max_j L(j)$
```

6.3 Edit distance



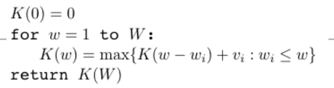
$$E(i,j) \quad = \quad \min\{1 + E(i-1,j),\ 1 + E(i,j-1),\ \mathbf{diff}(i,j) + E(i-1,j-1)\}$$

6.4 Knapsack



$K(w)$ = maximum value achievable with a knapsack of capacity w.

$$K(w) = \max_{i: w_i \le w} \{K(w - w_i) + v_i\}$$



```
$K(0) = 0$
for $w = 1$ to $W$:
    $K(w) = \max\{K(w - w_i) + v_i : w_i \le w\}$
return $K(W)$
```