

---

**Algorithm 1** Job scheduling

---

```
1: Sort the jobs in non-increasing order of profits:  $g_1 \geq g_2 \geq \dots \geq g_n$ 
2:  $d \leftarrow \max_i d_i$ 
3: for  $t : 1..d$  do
4:    $S(t) \leftarrow 0$ 
5: end for
6: for  $i : 1..n$  do
7:   Find the largest  $t$  such that  $S(t) = 0$  and  $t \leq d_i$ ,  $S(t) \leftarrow i$ 
8: end for
```

---

**Theorem 2.18.** *The greedy solution to job scheduling is optimal. That is, the profit  $P(S)$  of the schedule  $S$  computed by algorithm 1 is as large as possible.*

**Lemma 2.19.** *“ $S$  is promising” is an invariant for the (second) for-loop in algorithm 1.*

**Problem 2.21.** *Why does lemma 2.19 imply theorem 2.18? (Hint: this is a simple observation).*

*Proof.* Assume that after every iteration of the second for loop, the result is promising. After the final iteration,  $S$  is still promising, but the only unscheduled tasks are those that cannot extend  $S$  at any time. In other words,  $S$  is promising but it cannot be extended, so it must be optimal.  $\square$