19.3.1

Exercise: Scheduling Jobs to Minimize Weighted Average Waiting Time

Exercise: A Weighted Version

- n jobs $J_1, J_2, \ldots, J_n$. J_i has non-negative processing time p_i and a non-negative weight w_i
- One server/machine/person available to process jobs.
- Schedule/order the jobs to minimize total or average waiting time
- Waiting time of J_i in schedule σ: sum of processing times of all jobs scheduled before J_i
- Goal: minimize total weighted waiting time.
- Formally, compute a permutation π that minimizes $\sum_{i=1}^{n} \left(\sum_{j=1}^{i-1} p_{\pi(j)} \right) w_{\pi(i)}$.

	J_1	J_2	J_3	J_4	J_5	J_6
time	3	4	1	8	2	6
weight	10	5	2	100	1	1

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first
Pricing	$0 \cdot w_1 + p_1 w_2$	$0 w_2 + p_2 w_1$

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first
Pricing	$0 \cdot w_1 + p_1 w_2$	$0 w_2 + p_2 w_1$
Equivalent to	$p_1 w_2$	$p_2 w_1$

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first
Pricing	$0 \cdot w_1 + p_1 w_2$	$0 w_2 + p_2 w_1$
Equivalent to	$p_1 w_2$	$p_2 w_1$

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first
Pricing	$0 \cdot w_1 + p_1 w_2$	$0 w_2 + p_2 w_1$
Equivalent to	$p_1 w_2$	$p_2 w_1$

need to compare $p_1 w_2 \stackrel{?}{=} p_2 w_1$

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first
Pricing	$0 \cdot w_1 + p_1 w_2$	$0 w_2 + p_2 w_1$
Equivalent to	$p_1 w_2$	$p_2 w_1$

need to compare $p_1 w_2 \stackrel{?}{=} p_2 w_1$

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first
Pricing	$0 \cdot w_1 + p_1 w_2$	$0 w_2 + p_2 w_1$
Equivalent to	$p_1 w_2$	$p_2 w_1$

---

need to compare $p_1 w_2 \stackrel{?}{=} p_2 w_1$
dividing by $p_1 p_2$...

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first
Pricing	$0 \cdot w_1 + p_1 w_2$	$0 w_2 + p_2 w_1$
Equivalent to	$p_1 w_2$	$p_2 w_1$

---

need to compare $p_1 w_2 \stackrel{?}{=} p_2 w_1$
dividing by $p_1 p_2$...

equivalent to comparing $w_2 / p_2 \stackrel{?}{=} w_1 / p_1$
$\omega_i = w_i / p_i$: Price per processing unit in dollars

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first
Pricing	$0 \cdot w_1 + p_1 w_2$	$0 w_2 + p_2 w_1$
Equivalent to	$p_1 w_2$	$p_2 w_1$

---

need to compare $p_1 w_2 \stackrel{?}{=} p_2 w_1$
dividing by $p_1 p_2$...

equivalent to comparing $w_2 / p_2 \stackrel{?}{=} w_1 / p_1$
$\omega_i = w_i / p_i$: Price per processing unit in dollars

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first
Pricing	$0 \cdot w_1 + p_1 w_2$	$0 w_2 + p_2 w_1$
Equivalent to	$p_1 w_2$	$p_2 w_1$

---

need to compare $p_1 w_2 \stackrel{?}{=} p_2 w_1$
dividing by $p_1 p_2$...

equivalent to comparing $w_2 / p_2 \stackrel{?}{=} w_1 / p_1$
$\omega_i = w_i / p_i$: Price per processing unit in dollars
Sort jobs in decreasing value of ω_i. Schedule jobs by this value.

Exercise: A Weighted Version

Consider two jobs p_1, p_2 of weight w_1 and w_2. We have two possibilities:

	Job 1 first	Job 2 first
Pricing	$0 \cdot w_1 + p_1 w_2$	$0 w_2 + p_2 w_1$
Equivalent to	$p_1 w_2$	$p_2 w_1$

---

need to compare $p_1 w_2 \stackrel{?}{=} p_2 w_1$
dividing by $p_1 p_2$...

equivalent to comparing $w_2 / p_2 \stackrel{?}{=} w_1 / p_1$
$\omega_i = w_i / p_i$: Price per processing unit in dollars
Sort jobs in decreasing value of ω_i. Schedule jobs by this value.

Correctness proof: Same as the unweighted case – if there is an inversion, then by the argument above, flip these jobs, and get a better schedule.

THE END

...

(for now)