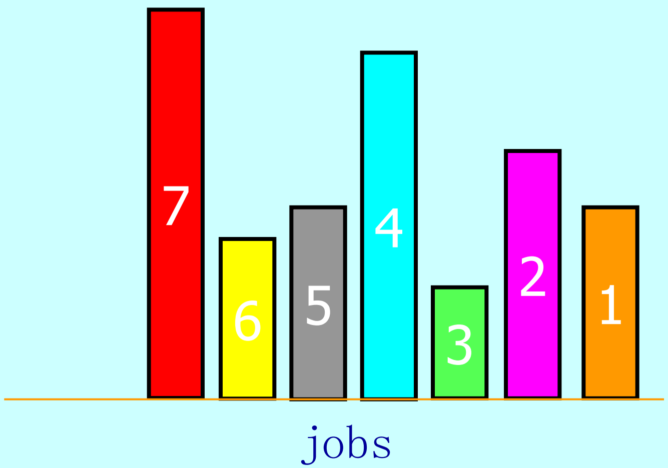
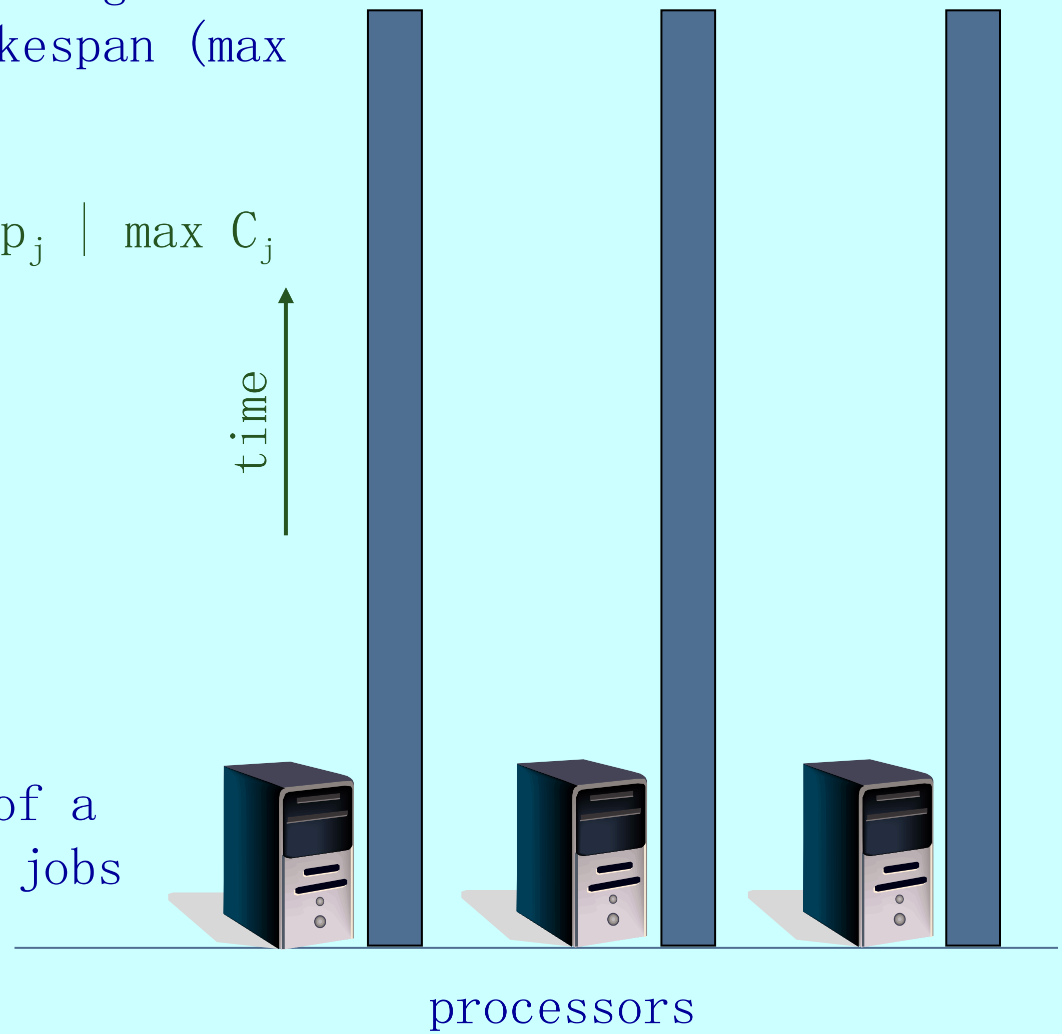
List Scheduling

Given a list of jobs (each with a specified processing time), assign them to processors to minimize makespan (max load)

In Graham's notation: $Pm \mid p_j \mid \max C_j$

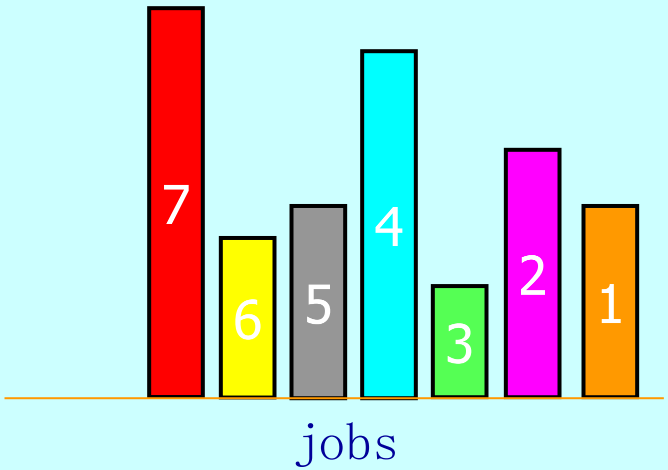


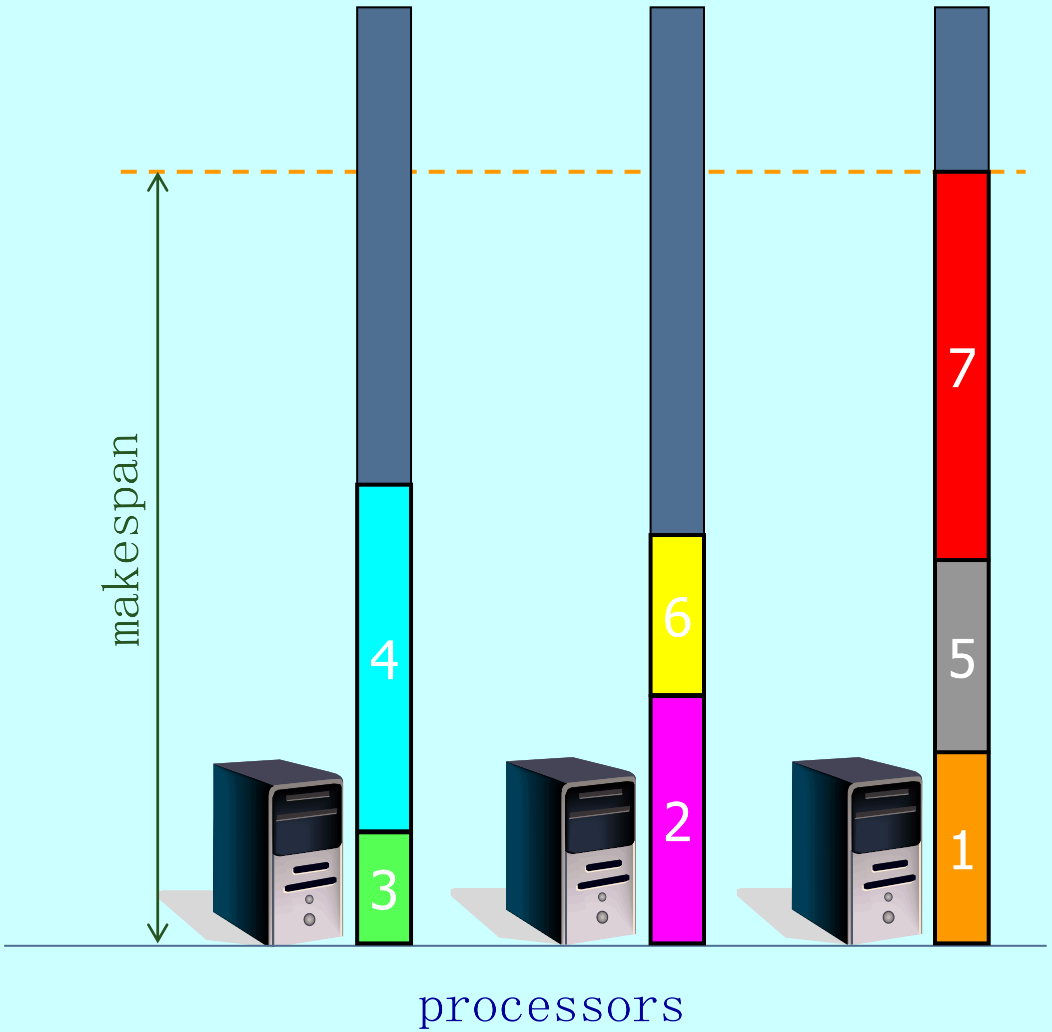


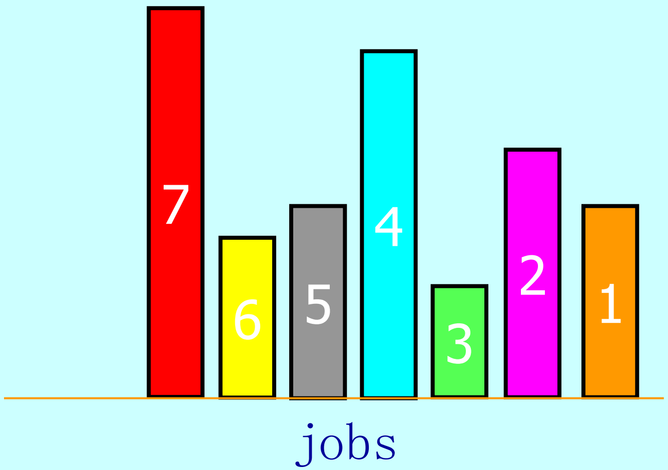
Online algorithm: assignment of a job does not depend on future jobs

Goal: small competitive ratio

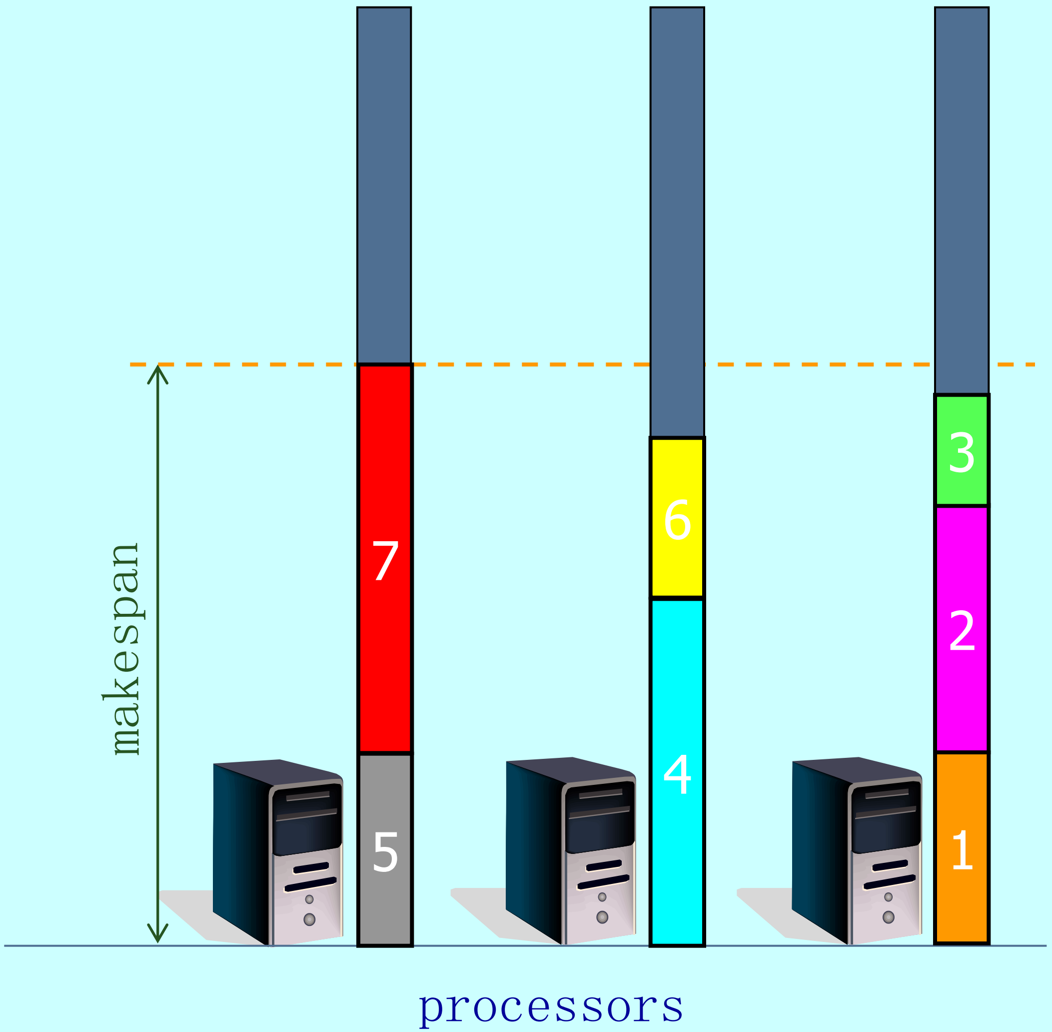
Greedy: Assign each job to the machine with the lightest load





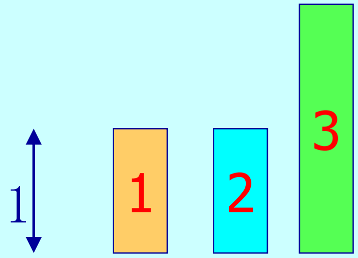


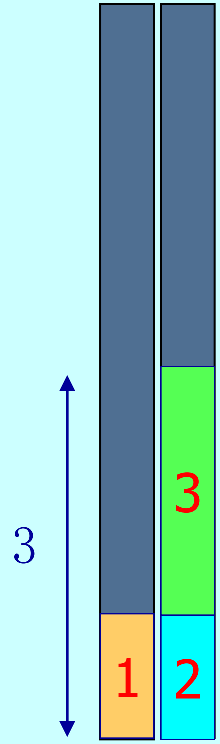
better schedule:



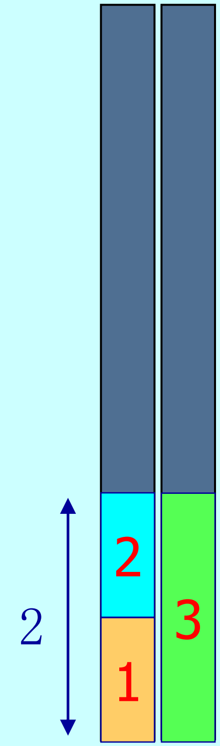
How bad is Greedy?

Try k = 2 first …





Greedy

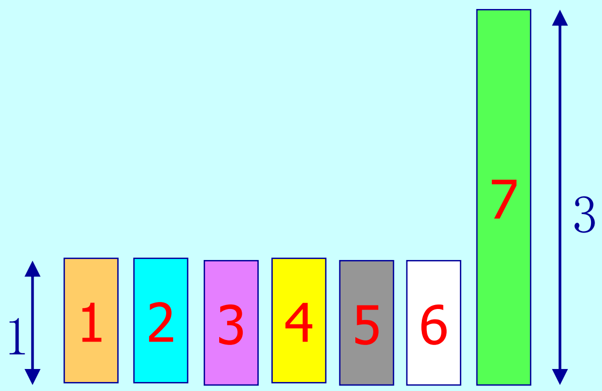


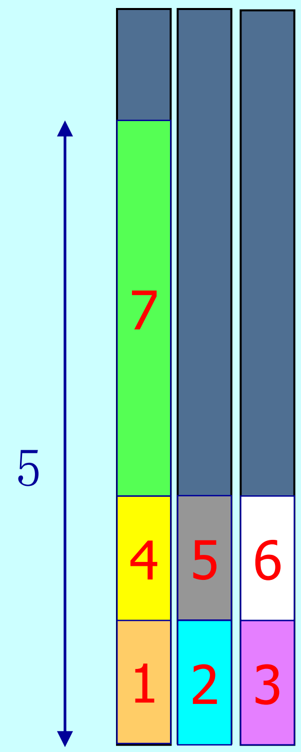
optimal

So for k = 2 Greedy’s competitive ratio is $\geq 3/2$

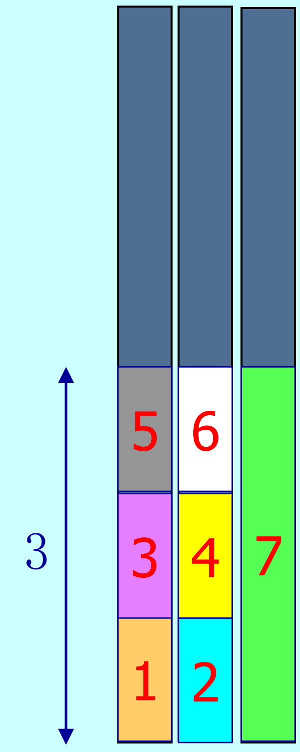
How bad is Greedy?

Try k = 3 now …





Greedy



optimal

So for m = 3 Greedy's competitive ratio is $\geq 5/3$

Exercise: Show that for m machines Greedy's competitive ratio is $\geq 2 - 1/m$

How good is Greedy?

Hint:
load ≥ $m \cdot x + y$

Rough analysis:

x = min load before placing last job
y = length of last job

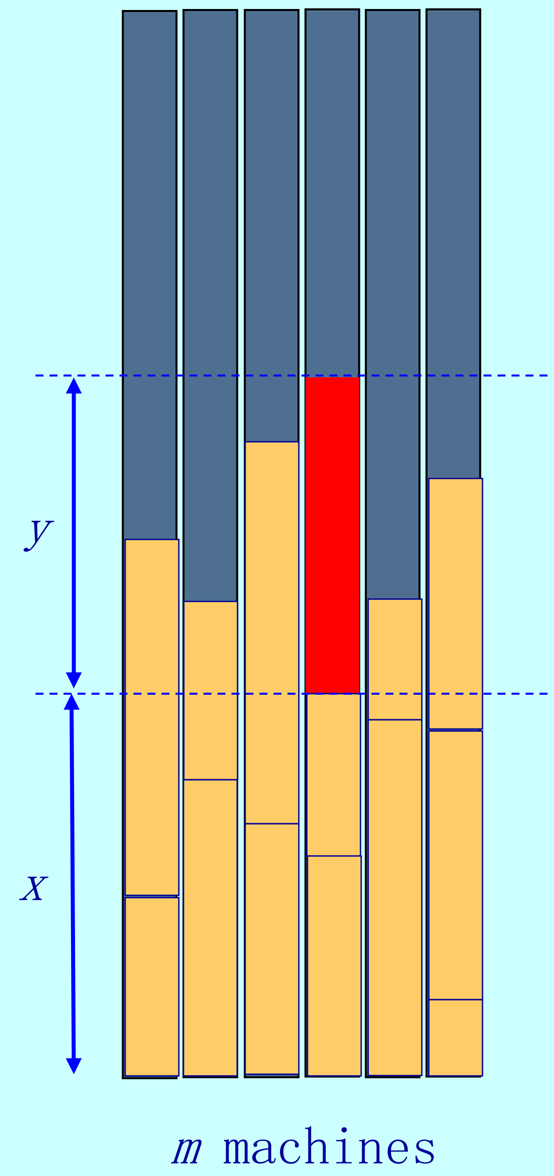
- total load ≥ $m \cdot x$, so optimum makespan ≥ x
- optimum makespan ≥ y

- so

greedy's makespan = $x + y$
≤ $2 \cdot \max(x, y)$
≤ $2 \cdot$ optimum

makespan

Exercise: Improve the bound to 2-1/m.



How good is Greedy?

- total load $\geq m \cdot x + y$,
  so optimum makespan $\geq x + y/m$
- optimum makespan $\geq y$

Find smallest R such that

$$x + y \leq R \cdot \max(y, x + y/m)$$

We can assume x+y = 1 and x, y ≥ 0

Then

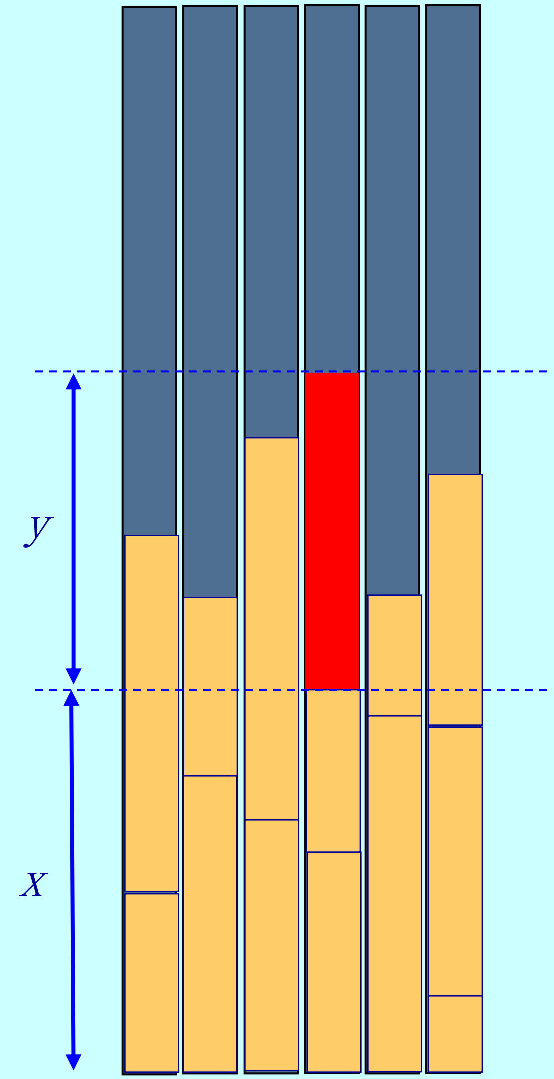
$$R = \max_y 1/\max(y, 1-y+y/m)$$

Equalizing, $y = 1-y+y/m$, we get

$$y = m/(2m-1)$$

Substituting into R, we get:

Theorem: Greedy is $(2-1/m)$-competitive.



m machines

List Scheduling

- Greedy is $(2-1/m)$-competitive [Graham '66]
- Lower bound ≈1.88
  [Rudin III, Chandrasekaran'03]
- Best known ratio ≈1.92
  [Albers '99] [Fleischer, Wahl '00]
- Lots of work on randomized algorithms, preemptive scheduling, …