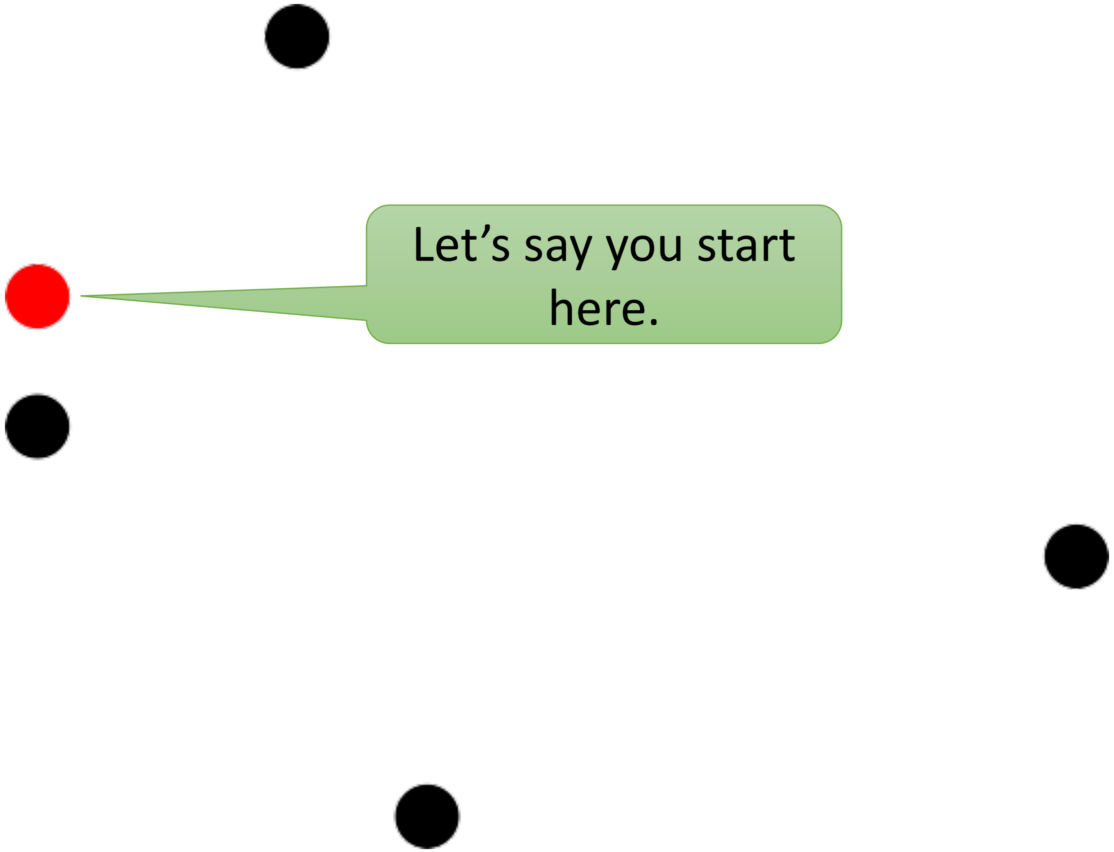
Competitive Programming

Greedy Algorithms

Greedy Choices

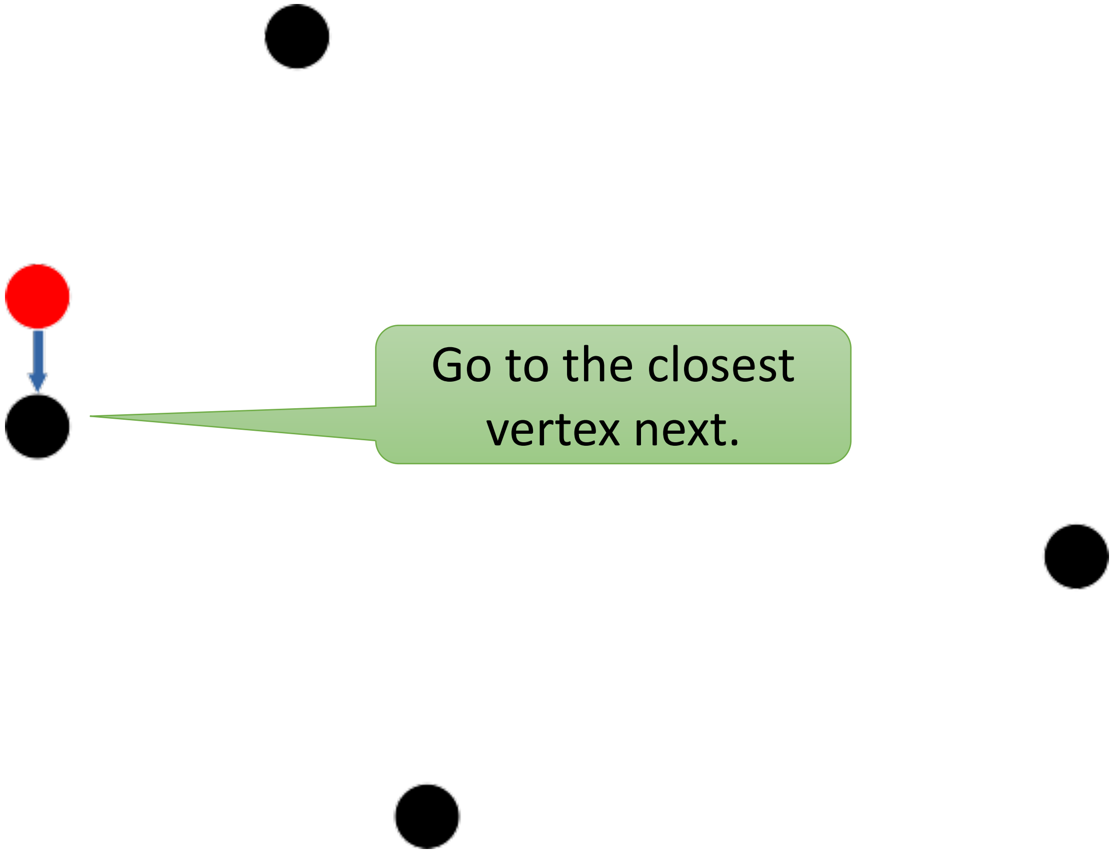
- Often, locally greedy choices don't yield globally optimal solutions.
  - TSP for example



Let's say you start here.

Greedy Choices

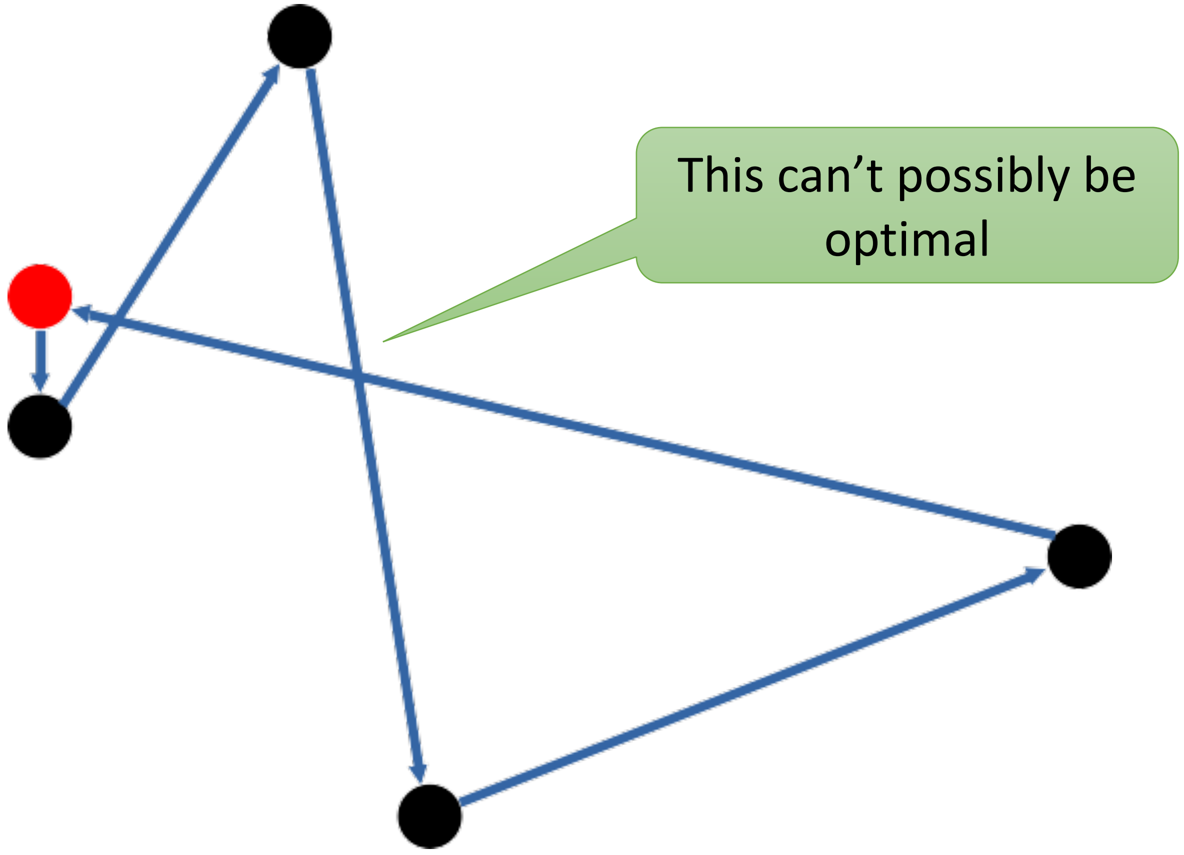
- Often, locally greedy choices don’t yield globally optimal solutions.
  - TSP for example



Go to the closest vertex next.

Greedy Choices

- Often, locally greedy choices don’t yield globally optimal solutions.
    - TSP for example
    - Any other NP Complete problem



This can’t possibly be optimal

Greedy Choices

- Often, locally greedy choices don’t yield globally optimal solutions.
  - TSP for example
  - Any other NP Complete problem
- For some problems, making a locally greedy choice leads to a globally optimal solution.
  - This is a little uncommon
  - You’d have to convince yourself that this really works.

Greedy Choices

- Often, locally greedy choices don’t yield globally optimal solutions.
    - TSP for example
    - Any other NP Complete problem
- For some problems, making a locally greedy choice leads to a globally optimal solution.
    - This is a little uncommon
    - You’d have to convince yourself that this really works.
    - That argument usually goes like:
      If you had an optimal solution …

Greedy Choices

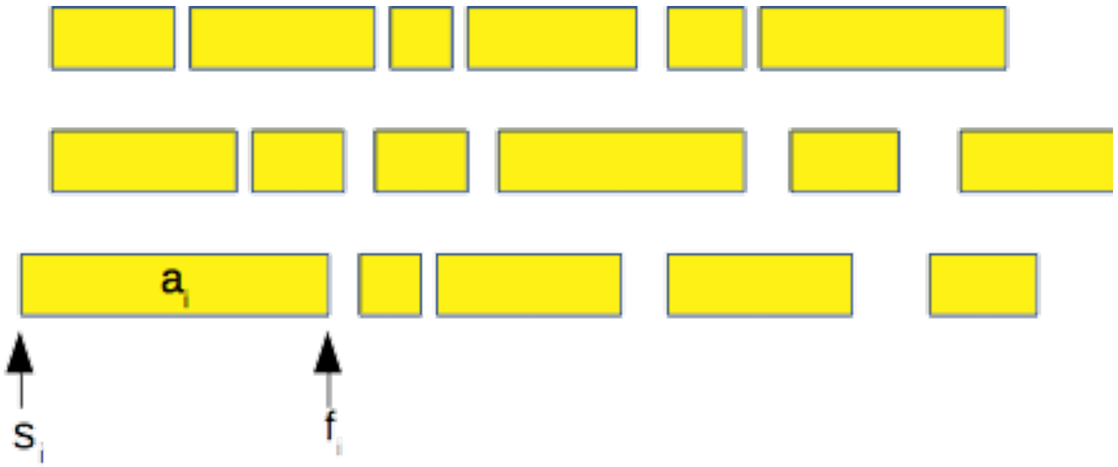
- Often, locally greedy choices don't yield globally optimal solutions.
    - TSP for example
    - Any other NP Complete problem
- For some problems, making a locally greedy choice leads to a globally optimal solution.
    - This is a little uncommon
    - You'd have to convince yourself that this really works.
    - That argument usually goes like:
      If you had an optimal solution …
      and it didn't have the greed choice I'm making here …

Greedy Choices

- Often, locally greedy choices don’t yield globally optimal solutions.
  - TSP for example
  - Any other NP Complete problem
- For some problems, making a locally greedy choice leads to a globally optimal solution.
  - This is a little uncommon
  - You’d have to convince yourself that this really works.
  - That argument usually goes like:
    If you had an optimal solution …
    and it didn’t have the greed choice I’m making here …
    I would be able to improve your solution (or do just as well)

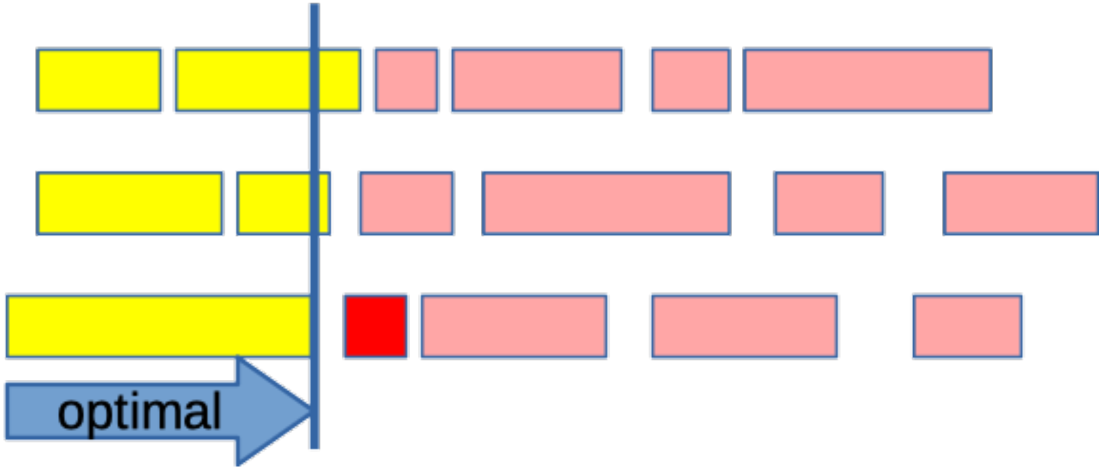
Activity Scheduling

- Given a set of n activities.
- Each with a starting time, s_1 .. s_n
- And a finishing time, f_1 .. f_n
- Want to choose as many activities as possible to participate in



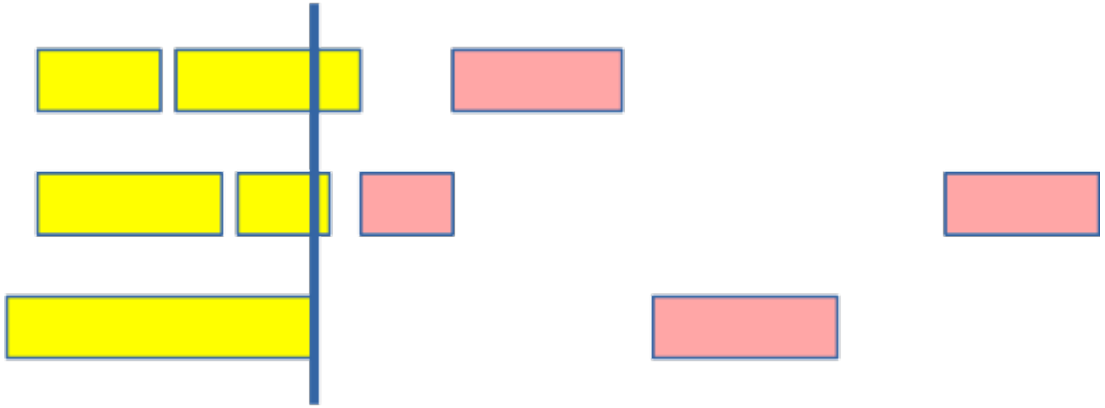
Activity Scheduling

- You can always make a greedy choice
  the next available event that finishes earliest.



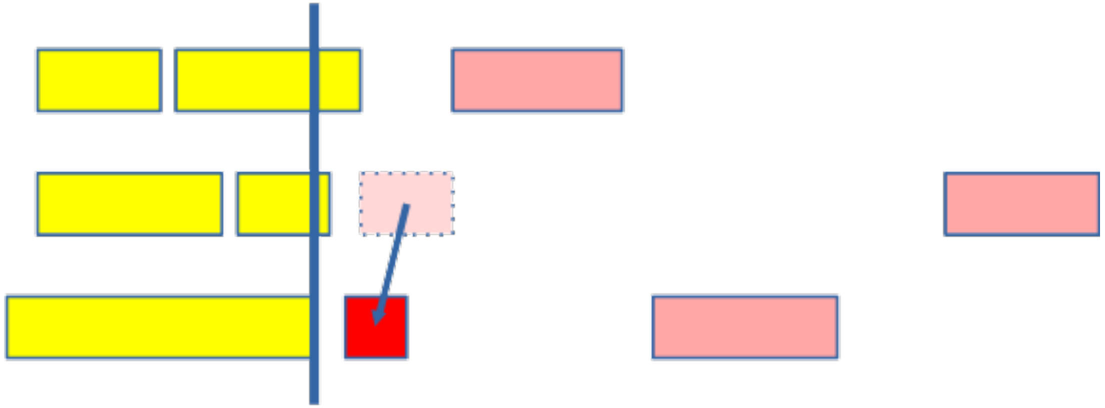
Activity Scheduling

- If there’s some optimal solution that doesn’t use that event.



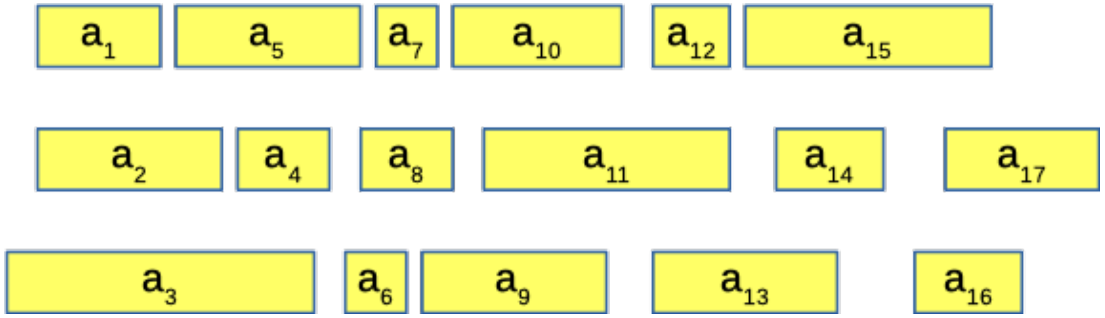
Activity Scheduling

- You can do at least as well by exchanging the next event for the next event for the greedy-choice event.



Activity Scheduling

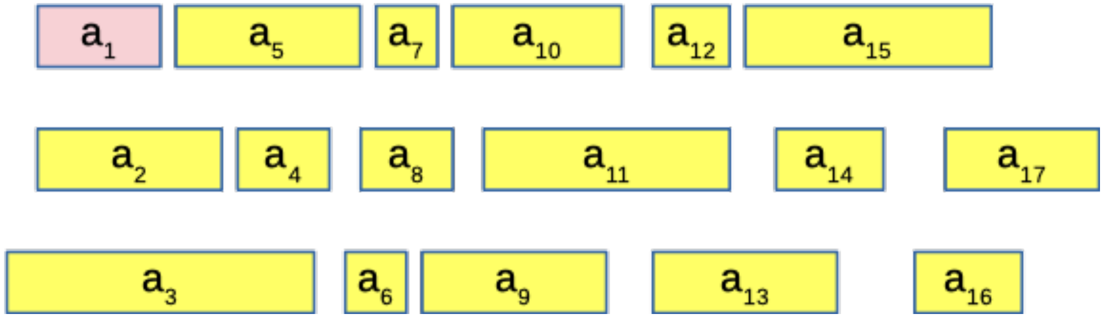
- To make this easy, we can first sort by finishing time.



Activity Scheduling

Sort by finishing time

Take the first-finishing event

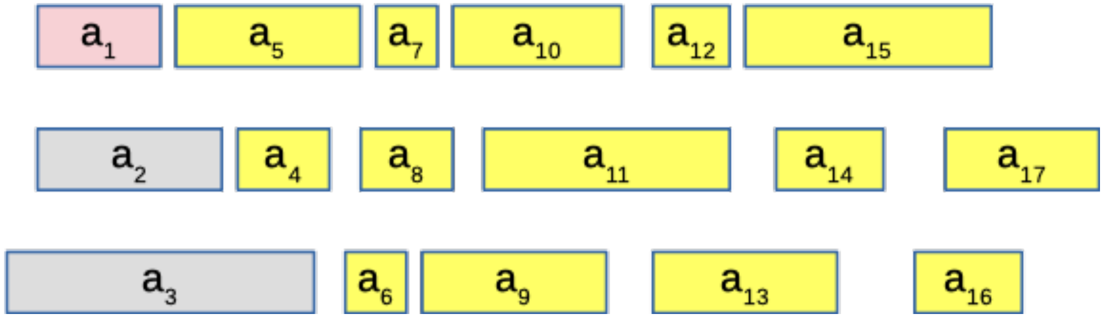


Activity Scheduling

Take the first-finishing event

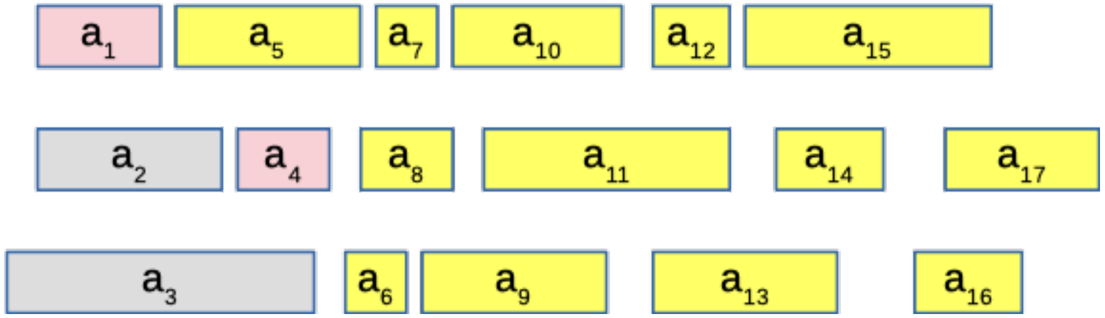
Repeat until no more events

- Discard events that start too early to use



Activity Scheduling

- Take the first-finishing event
- Repeat until no more events
  - Discard events that start too early to use
  - Take the next event

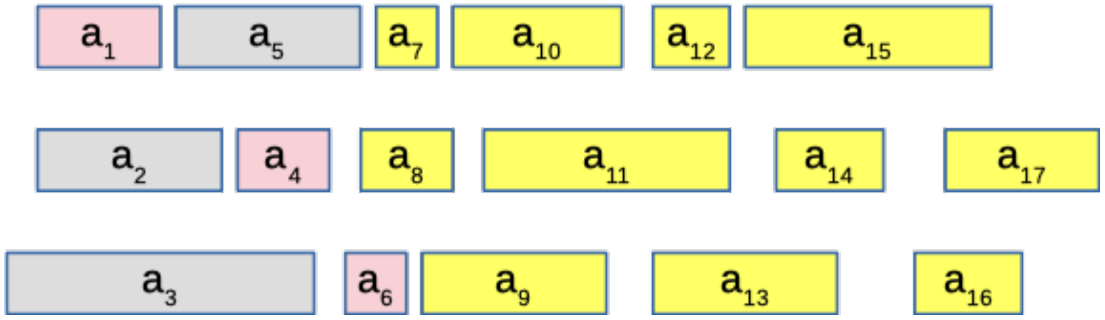


Activity Scheduling

Take the first-finishing event

Repeat until no more events

- Discard events that start too early to use
- Take the next event

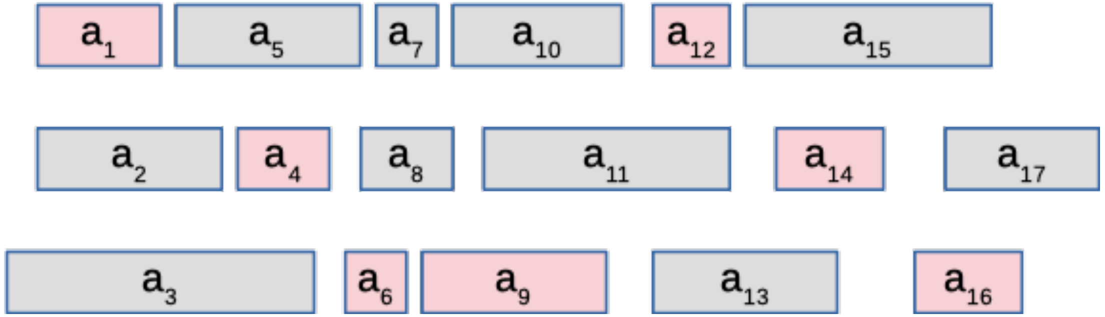


Activity Scheduling

Take the first-finishing event

Repeat until no more events

- Discard events that start too early to use
- Take the next event


a_1
a_5
a_7
a_{10}
a_{12}
a_{15}
a_2
a_4
a_8
a_{11}
a_{14}
a_{17}
a_3
a_6
a_9
a_{13}
a_{16}