Approximation Algorithms

Approximation Algorithms

- Formal definition says,

  "Given an optimization problem ***P****, an algorithm* ***A*** *is said to be an approximate algorithm for* ***P****, if for any given instance* ***I****, it returns an approximate solution i.e. a feasible solution."*

Approximation Algorithms

- An approximation algorithm is a way of dealing with NP-completeness for an optimization problem.
- It is used when finding optimal solution for a NP-Hard problem.
- Majority of optimization problem comes under NP-Hard or NP-Complete.
- This technique does not guarantee the best solution.

Approximation Algorithms

- The goal of the approximation algorithm is to come as close as possible to the optimal solution in polynomial time.
- Such algorithms are called approximation algorithms or heuristic algorithms.

- For the traveling salesperson problem, the optimization problem is to find the shortest cycle, and the approximation problem is to find a short cycle.
- For the vertex cover problem, the optimization problem is to find the vertex cover with fewest vertices, and the approximation problem is to find the vertex cover with few vertices.

Approximation Algorithms - Features

- An approximation algorithm guarantees to run in polynomial time though it does not guarantee the most effective solution.
- An approximation algorithm guarantees to seek out high accuracy and top quality solution(say within 1% of optimum)
- Approximation algorithms are used to get an answer near the (optimal) solution of an optimization problem in polynomial time

The Bin Packing Problem

- Bin Packing Problem is a famous problem of Approximation Algorithms

Problem Statement

- Given **n** items of different weights and bins each of capacity **c**, assign each item to a bin such that number of total used bins is minimized.
- Ie, objective is to find the minimum number of bins needed to accommodate n items.
- It may be assumed that all items have weights smaller than bin capacity.
- No objects can be placed partially in one bin and rest of portion in other bin.

The Bin Packing Problem

- Four simple solutions are:

  a) First fit

  b)Best fit

  c) First fit decreasing

  d) Best fit decreasing.

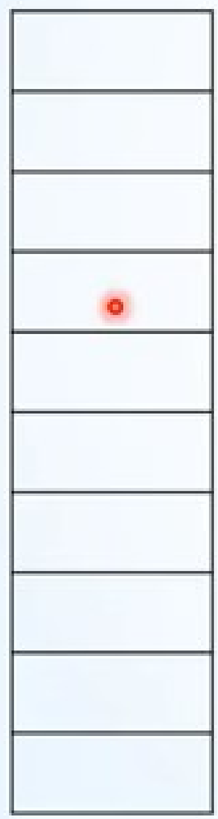
First fit

- Index the bins 1, 2, 3…..
- All bins are initially filled to level 0
- Objects are considered for packing in the order 1, 2, 3, …..n
- To pack object i find the least index j such that bin j is filled to a level r, $r \le L - l_i$
- Pack i into bin j .
- Bin j is now filled to level $r + l_i$

First fit

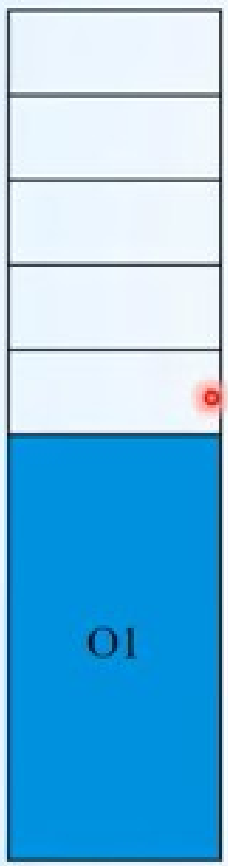
- L is capacity of bin an n is no.of objects.

- Let L = 10, n = 6 (l_1, l_2, l_3, l_4, l_5, l_6,) = (5, 6, 3, 7, 5, 4)



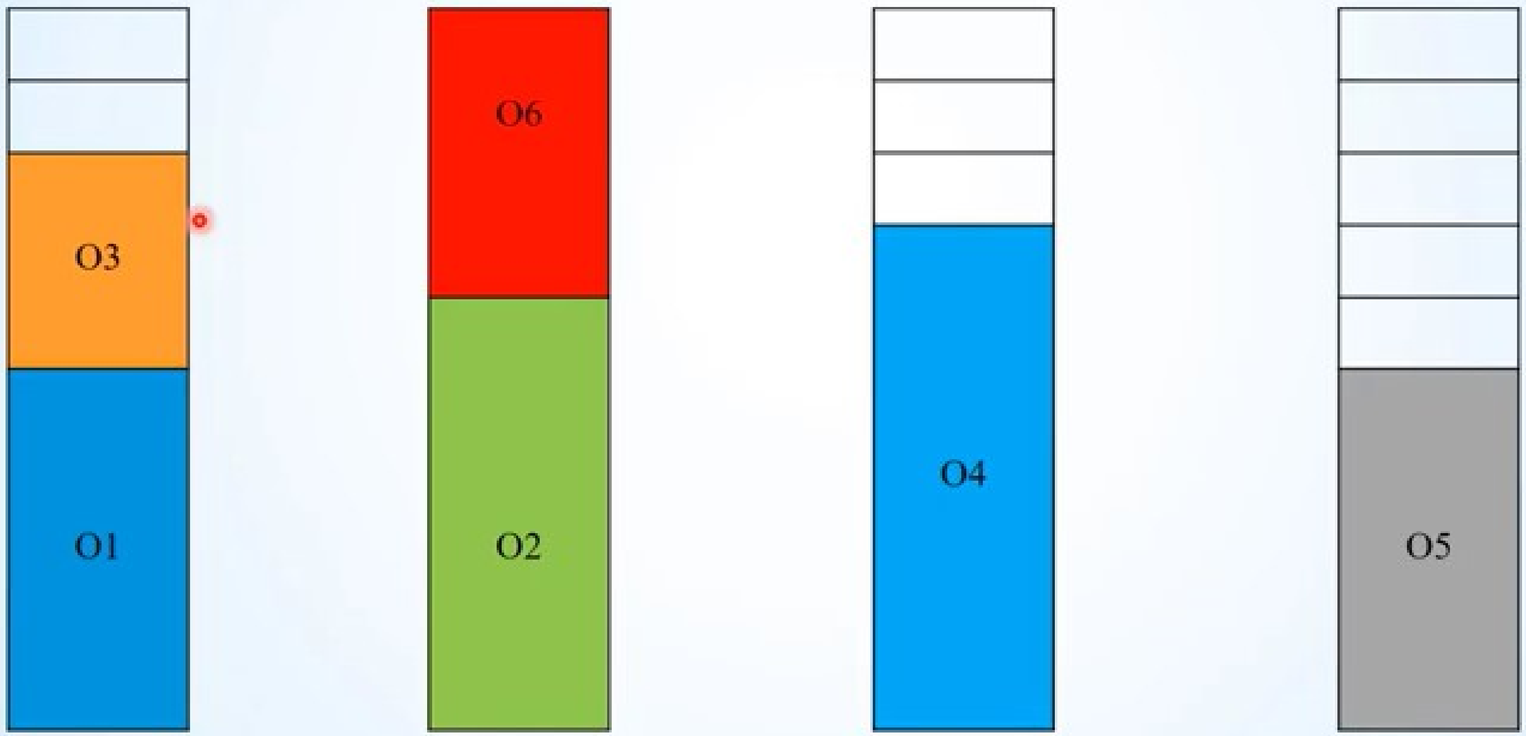
First fit

- Let L = 10, n = 6 ($\mathbf{l_1}$, l_2, l_3, l_4, l_5, l_6,) = (**5**, 6, 3, 7, 5, 4)



First fit

- Let $L = 10$, $n = 6$ $(l_1, l_2, l_3, l_4, l_5, \mathbf{l_6})$ = (5, 6, 3, 7, 5, **4**)
- **Total no: of bins = 4**

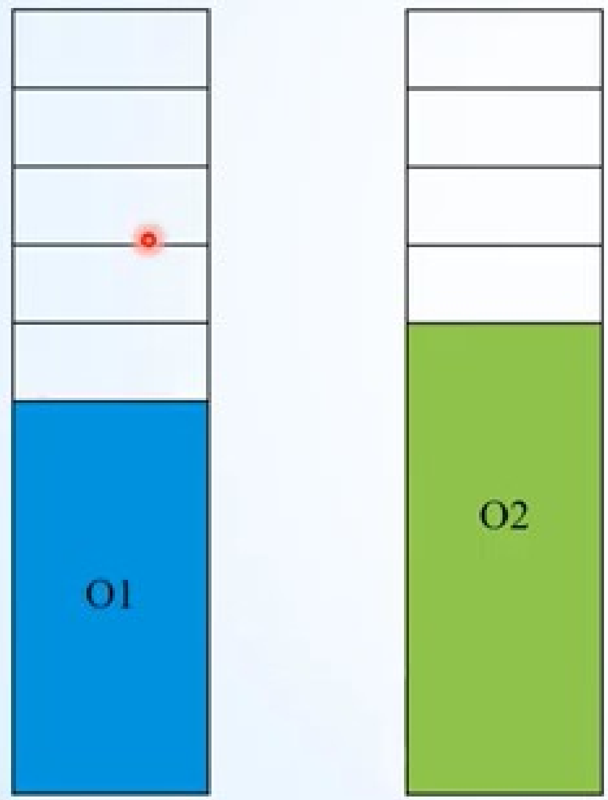


Best fit

- Initial condition is same as first fit.
- When object I is being considered find the least j such that bin j is filled to a level r, $r<=L-l_i$ and as large as possible.
- Pack I into bin j.
- Bin j is now filled to level $r+l_i$

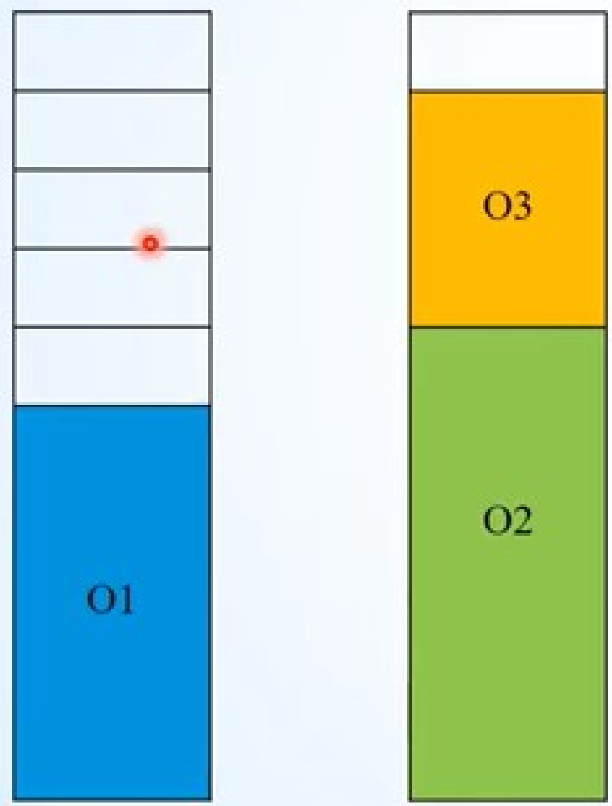
Best fit

- Let $L = 10$, $n = 6$ $(l_1, l_2, l_3, l_4, l_5, l_6,\ ) = (5, 6, 3, 7, 5, 4)$



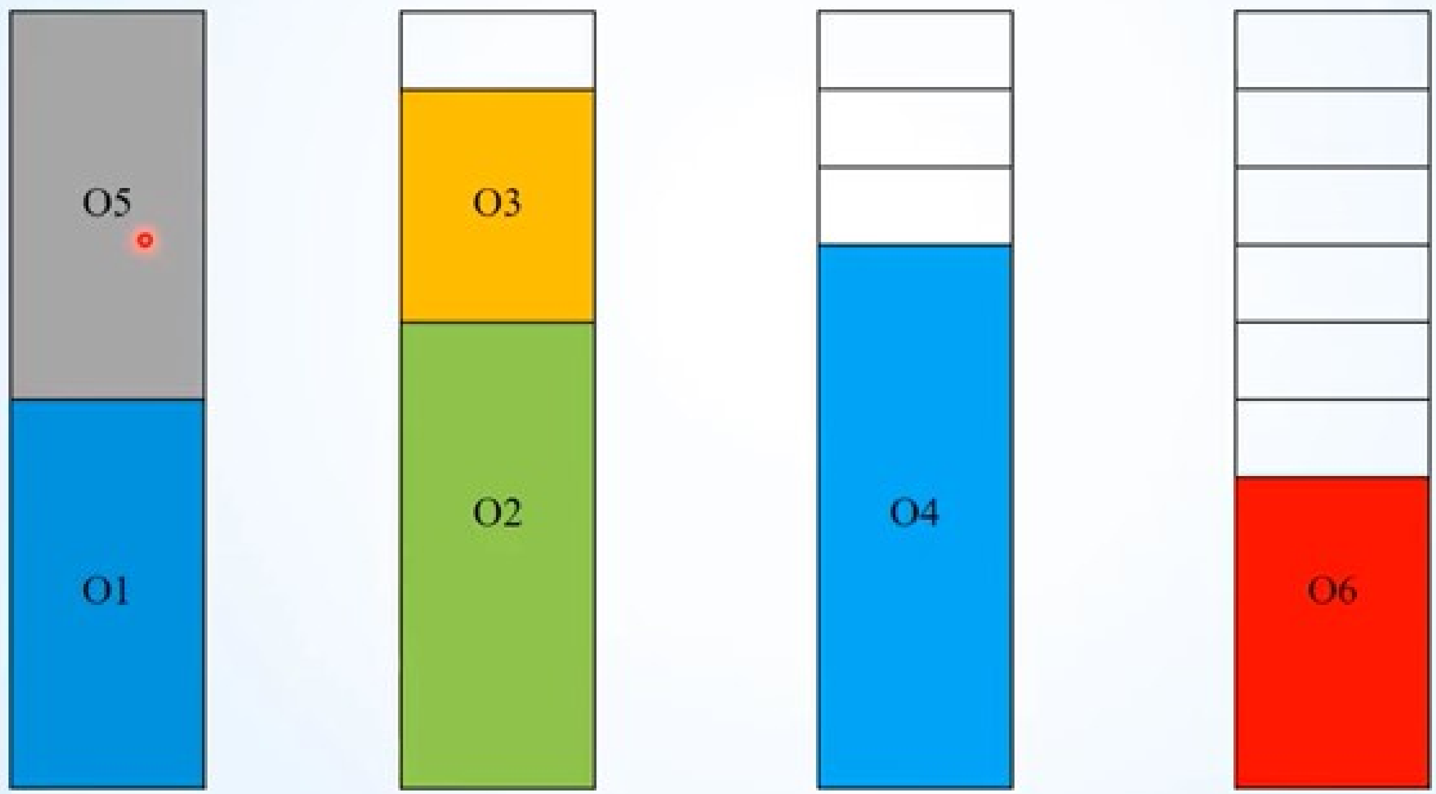
Best fit

- Let $L = 10$, $n = 6$ $(l_1, l_2, l_3, l_4, l_5, l_6,\ ) = (5, 6, 3, 7, 5, 4)$



Best fit

- Let $L = 10$, $n = 6$ $(l_1, l_2, l_3, l_4, l_5, l_6,) = (5, 6, 3, 7, 5, 4)$
- No :of bins = 4



First fit decreasing

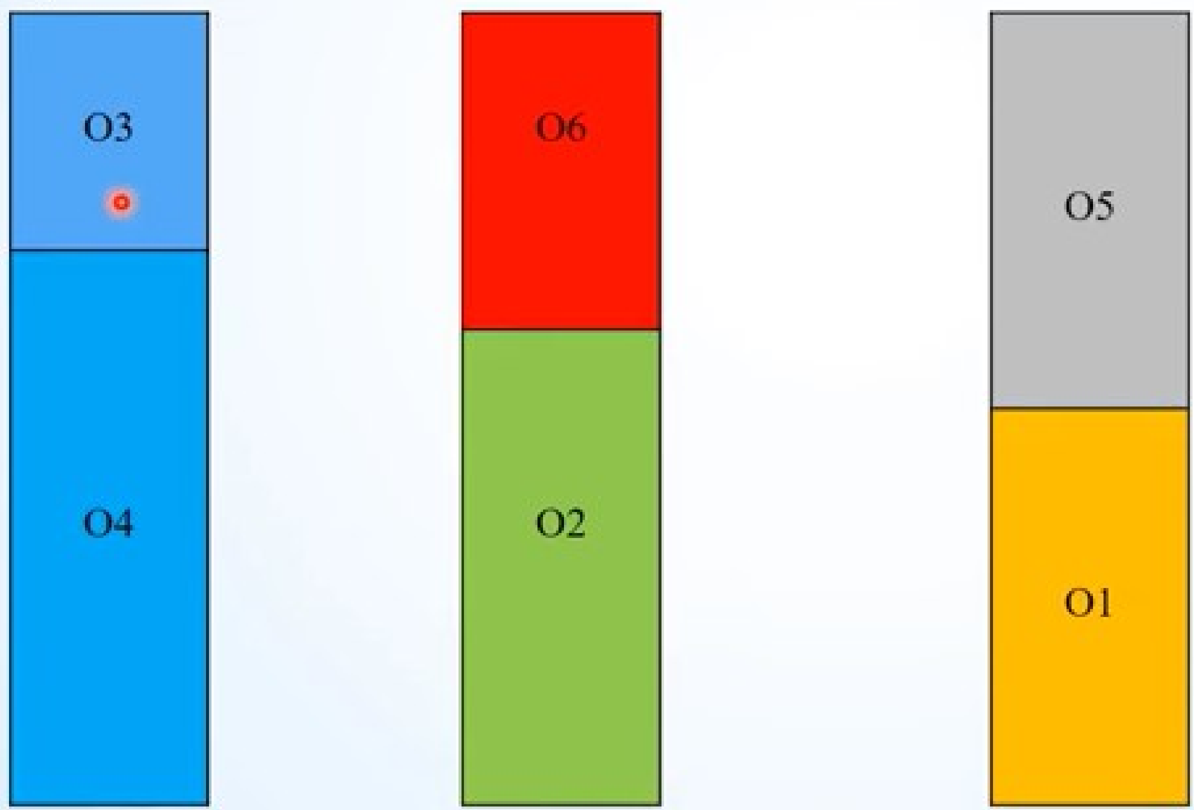
- Reorder the objects so that $l_i \geq l_i + 1,\ 1 \leq\ i \leq n$
- Now First Fit to pack the objects

First fit decreasing

- Let $L = 10$, $n = 6$ $(l_1, l_2, l_3, l_4, l_5, l_6,) = (5, 6, 3, 7, 5, 4)$
- $(l_4, l_2, l_1, l_5, l_6, l_3,) = (7, 6, 5, 5, 4, 3)$

First fit decreasing

- Let $L = 10$, $n = 6$ $(l_1, l_2, l_3, l_4, l_5, l_6,) = (5, 6, 3, 7, 5, 4)$
- $(l_4, l_2, l_1, l_5, l_6, \mathbf{l_3},) = (7, 6, 5, 5, 4, \mathbf{3})$
- No :of Bins = 3



Best fit decreasing

- Reorder the objects so that $l_i \geq l_i + 1, 1 \leq i \leq n$
- Now Best fit to pack the objects.
- In our example it executes as same as First Fit Decreasing
- So No :of bins = 3
- Although First Fit Decreasing and Best Fit Decreasing obtain optimal packing on this example they do not in general obtain such packing

The Bin Packing Problem

- Here in this example we got optimal solution.
- But it is not sure to get an optimal solution.
- Ie, Approximation algorithms gives nearest optimal solution.