Binary Search

Sarnen Camp 2019

February 14, 2019

Binary Search

```
// Input: [0,0,...,0,1,...,1,1]
// Output: Index of last 0, or -1 if there is none
int search(vector<int> const& a) {
  // Invariant: l==-1 or a[l]=0
  int l=-1, r=a.size();
  while (r - l > 1) {
    int m = l + (r - l)/2;
    if (a[m] == 0)
      l = m;
    else
      r = m;
  }
  return l;
}
```

Remarks

- Make sure that the list you are binary searching is sorted.
- You can also use a custom comparing operator.
- Set the pointers *l* and *r* outside of the list as shown in the code before.

Example 1

N employees

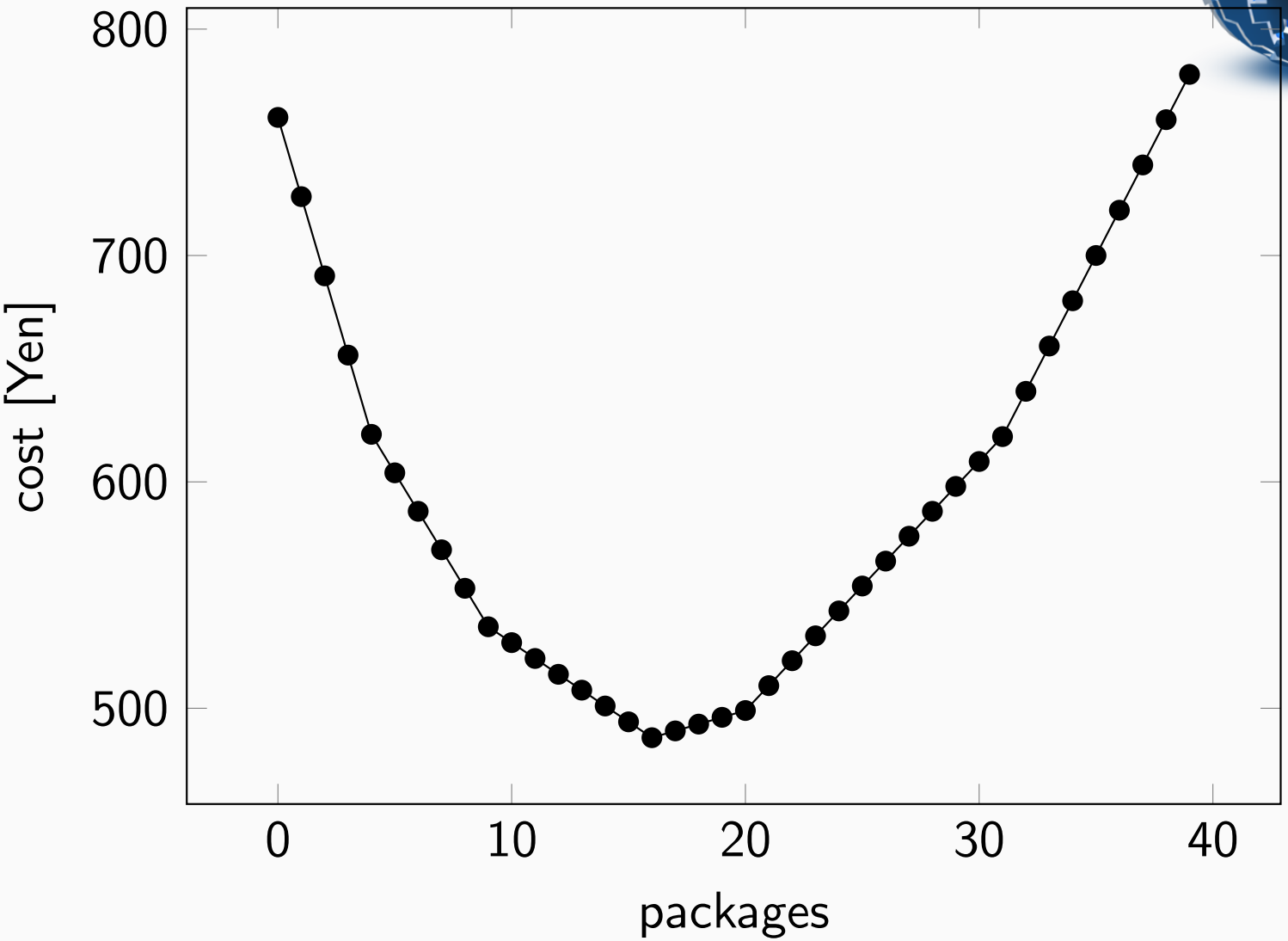
K filing cabinets that contain $a_0 ..., a_{k-1}$ documents

Assign employee i to consecutive cabinets $l_i, ..., r_i$

All cabinets need to be covered by exactly one employee

Minimize $\max_i \{a_{l_i} + a_{l_i+1} + \cdots + a_{r_i-1}\}$

Search on Convex Function



Search on Convex Function

- Binary search on derivative
- Ternary search (if you can't compute the derivative)

Example 2

- $n \cdot n$ grid of farmland.
- h_{ij} the height of a single field
- A the minimal number of connected fields
- Search for the lowest possible $\max h_{ij}$ with at least A connected fields.

Exponential Search

Sometimes, there is no upper bound or you want your solution to be in $\mathcal{O}(log(ans))$.

```
int r = 1;
while (!pred(r))
  r *= 2;
return search(l, r);
```

Binary Search Tree

Can visualize binary search as a tree.