顺序统计量

```
1 最大值
2 最小值
3 中位数
```

同时选最大最小

> 分奇偶
> 两两一对，内部先比较
> 奇：把初始元素自己变成2元对
> 新2元对与已有的最大最小对两次比较，更新最大最小2元对

- 时间3[n/2](下取整)

找第i大

> good partition：把规模降到0.8n

- 巧妙定义随机变量Ci：

从最后一次good到第i次good需要的时间

RndSelect(A, i): **A Divide-and-Conquer Alg.**

```
if (A.size=1)
  return A[1]
else
  q = RandomPartition(A)
  if (i==q)
    return A[q]
  elseif (i<q)
    return RndSelect(A[1…(q-1)],i)
  else
    return RndSelect(A[(q+1)…A.size],i-q)
```

- 期望时间复杂度O（n）

$$\mathbb{E}[C_i] \le \Theta(1) \cdot 0.8^{i-1} n$$

find some o.s.

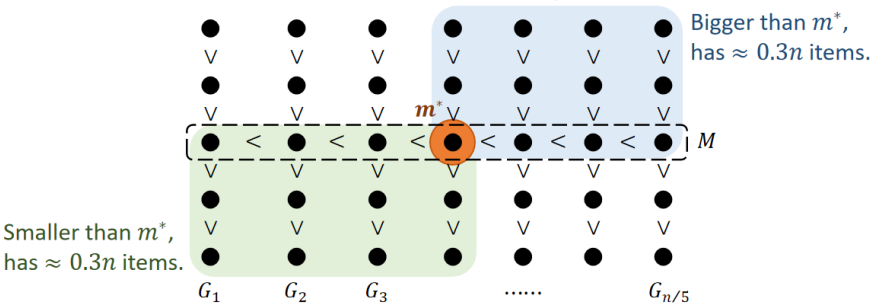
$$\mathbb{E}[T(n)] \le \mathbb{E}\left[\sum_{i=1}^{\log_{1.25} n} C_i\right] = \sum_{i=1}^{\log_{1.25} n} \mathbb{E}[C_i] = O(n)$$

we are done.

改进partition--m of ms

Median of medians

- Divide elements into $n/5$ groups, each containing 5 elements, call these groups $G_1, G_2, \cdots, G_{n/5}$.
- Find the medians of these $n/5$ groups, let M be this set of medians.
- Find the median of M, call it m^*. Partition using m^* as pivot is good: the smaller split has $\geq 0.3n$ items.



Time complexity?

MedianOfMedians(A):

```
<G1,G2,…,Gn/5> = CreateGroups(A)
for (i=1 to n/5)
  Sort(Gi)
M = GetMediansFromSortedGroups(G1,G2,…,Gn/5)
return QuickSelect(M,(n/5)/2)
```

QuickSelect(A, i):

```
if (A.size=1)
  return A[1]
else
  m = MedianOfMedians(A)
  q = PartitionWithPivot(A,m)
  if (i==q)
    return A[q]
  elseif (i<q)
    return QuickSelect(A[1…(q-1)],i)
  else
    return QuickSelect(A[(q+1)…A.size],i-q)
```

- 最坏线性时间复杂度

$$T(n) \leq T(0.7n) + T(0.2n) + O(n)$$

$$T(n) = O(n)$$