طراحی الگوریتم ها

۳۱ شهریور و ۲ مهر

ملکی مجد

معرفی درس

- mmalekimajd@gmail.com
- Room 307

- Telegram group

- References
  - Books!
  - Google
  - Prof. and TAs

CLRS	Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein
JeffE	Algorithms by Jeff Erickson

- نمره دهی
  - میان ترم و پایان ترم ۱۴. تمرین ها و پروژه ۴. مشارکت کلاسی ۲ (درصد نمره تمرین ها با توجه به تعداد آن ها ممکن است تغییر کند)

مقدمه

- الگوریتم
- اثبات درستی الگوریتم
- زمان اجرا
- روش های کارا برای حل مسئله

Topic	Reference
Recursion and Backtracking	Ch.1 and Ch.2 JeffE
Dynamic Programming	Ch.3 JeffE and Ch.15 CLRS
Greedy Algorithms	Ch.4 JeffE and Ch.16 CLRS
Amortized Analysis	Ch.17 CLRS
Elementary Graph algorithms	Ch.6 JeffE and Ch.22 CLRS
Minimum Spanning Trees	Ch.7 JeffE and Ch.23 CLRS
Single-Source Shortest Paths	Ch.8 JeffE and Ch.24 CLRS
All-Pairs Shortest Paths	Ch.9 JeffE and Ch.25 CLRS
Maximum Flow	Ch.10 JeffE and Ch.26 CLRS
String Matching	Ch.32 CLRS
NP-Completeness	Ch.12 JeffE and Ch.34 CLRS

بازگشت Recursion

- نمونه ای از کاهش
- برج هانوی
- مرتب سازی ادغامی
- مرتب سازی سریع
- الگوی تقسیم و حل
- درخت بازگشتی
- انتخاب در زمان خطی

انتخاب در زمان خطی

- میانه و ترتیب آماری
  - Medians and Order Statistics (فصل ۹ از کتاب CLRS)

- کمترین و بیشترین مقدار
- انتخاب تصادفی (تحلیل زمان مورد انتظار) Selection in expected linear time
- انتخاب در زمان خطی (تحلیل بدترین حالت)

انتخاب در زمان خطی

- میانه و ترتیب آماری
- Medians and Order Statistics (فصل ۹ از کتاب CLRS)

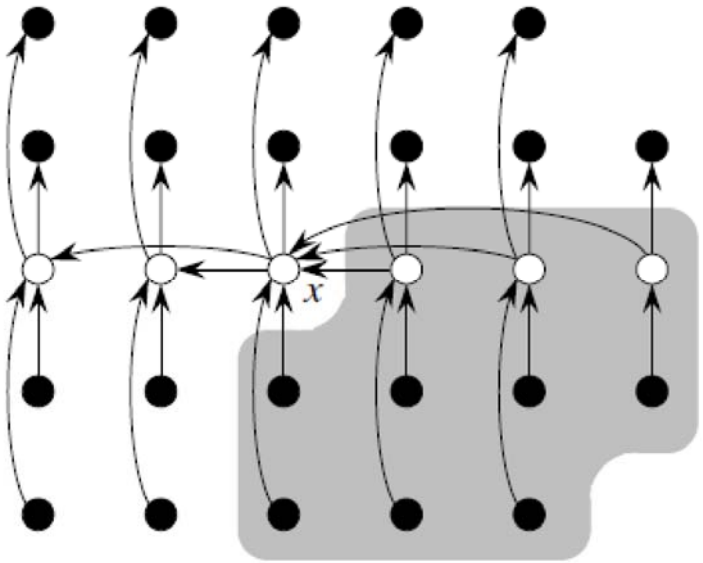
- کمترین و بیشترین مقدار
- انتخاب تصادفی (تحلیل زمان مورد انتظار) Selection in expected linear time
- انتخاب در زمان خطی (تحلیل بدترین حالت)

Input: A set A of n (distinct) numbers and a number i, with $1 \leq i \leq n$.

Output: The element $x \in A$ that is larger than exactly $i - 1$ other elements of A.

```
RANDOMIZED-SELECT(A, p, r, i)
1  if p = r
2     then return A[p]
3  q ← RANDOMIZED-PARTITION(A, p, r)
4  k ← q − p + 1
5  if i = k              ▷ the pivot value is the answer
6     then return A[q]
7  elseif i < k
8     then return RANDOMIZED-SELECT(A, p, q − 1, i)
9  else return RANDOMIZED-SELECT(A, q + 1, r, i − k)
```

1. Divide the n elements of the input array into $\lfloor n/5 \rfloor$ groups of 5 elements each and at most one group made up of the remaining $n \bmod 5$ elements.
2. Find the median of each of the $\lceil n/5 \rceil$ groups by first insertion sorting the elements of each group (of which there are at most 5) and then picking the median from the sorted list of group elements.
3. Use SELECT recursively to find the median x of the $\lceil n/5 \rceil$ medians found in step 2. (If there are an even number of medians, then by our convention, x is the lower median.)
4. Partition the input array around the median-of-medians x using the modified version of PARTITION. Let k be one more than the number of elements on the low side of the partition, so that x is the kth smallest element and there are $n-k$ elements on the high side of the partition.
5. If $i = k$, then return x. Otherwise, use SELECT recursively to find the ith smallest element on the low side if $i < k$, or the $(i-k)$th smallest element on the high side if $i > k$.



$$3\left(\left\lceil \frac{1}{2}\left\lceil \frac{n}{5}\right\rceil \right\rceil - 2\right) \geq \frac{3n}{10} - 6$$

$$T(n) \leq \begin{cases} \Theta(1) & \text{if } n \leq 140\,, \\ T(\lceil n/5 \rceil) + T(7n/10 + 6) + O(n) & \text{if } n > 140\,. \end{cases}$$

بازگشت به عقب

- مسئله n وزیر
- درخت بازی
- Subset sum
- طرح کلی
- تقسیم بندی متن (Text Segmentation)
- طولانی ترین توالی افزایش (Longest increasing subsequence)
- درخت بهینه جستجوی دودویی (Optimal binary search trees)

درخت بازی

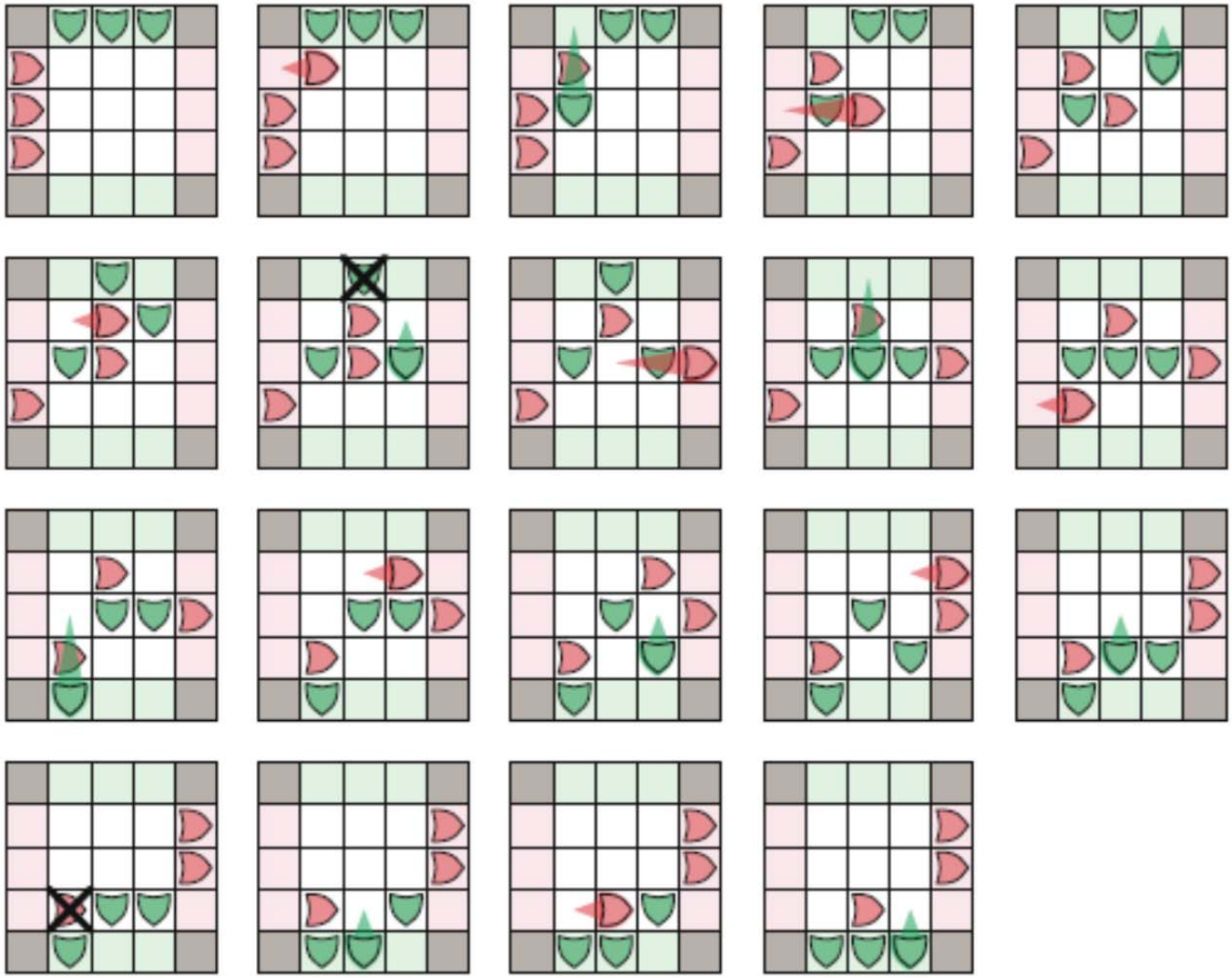


Figure 2.4. Vera wins the 3 × 3 fake-sugar-packet game.

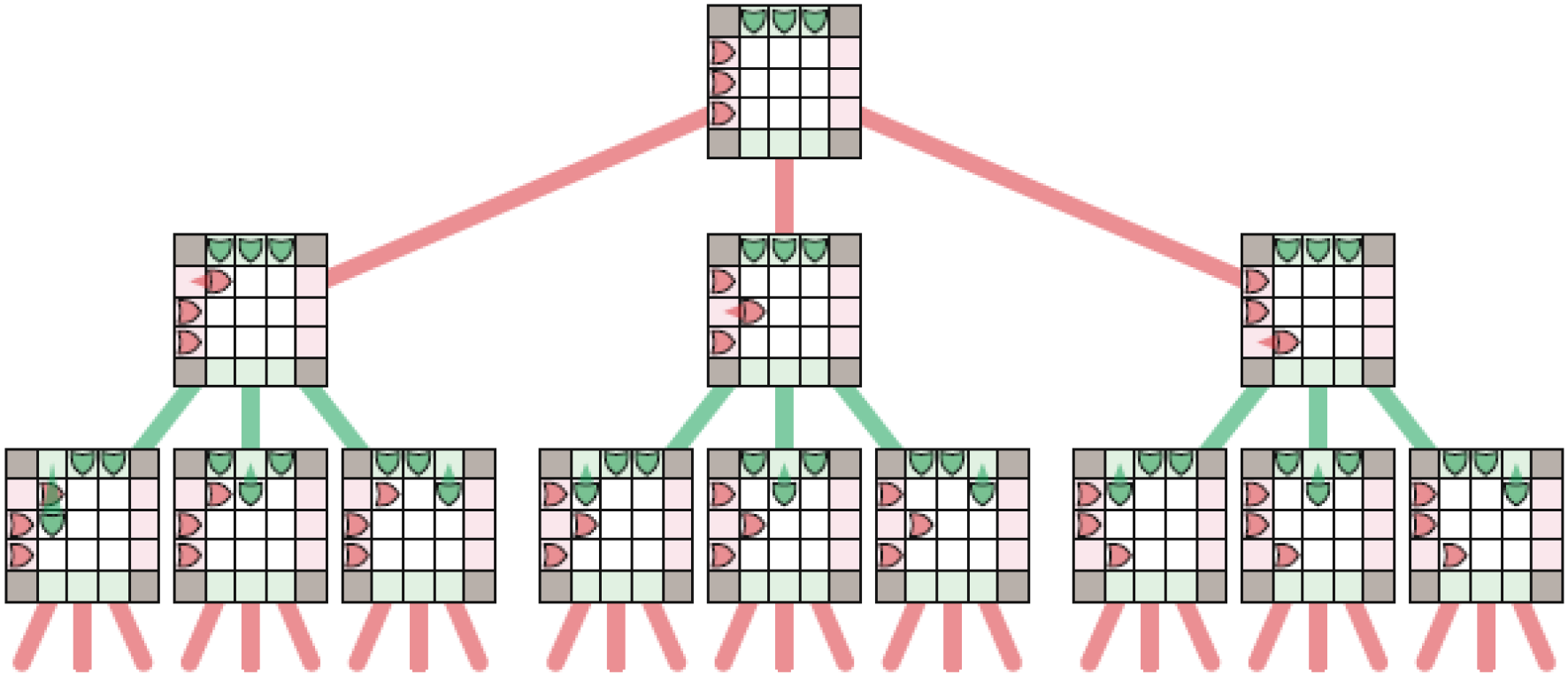


Figure 2.5. The first two levels of the fake-sugar-packet game tree.

```
PLAYANYGAME(X, player):
  if player has already won in state X
      return GOOD
  if player has already lost in state X
      return BAD
  for all legal moves X ⇝ Y
      if PLAYANYGAME(Y, ¬player) = BAD
          return GOOD        ⟨⟨X ⇝ Y is a good move⟩⟩
  return BAD                 ⟨⟨There are no good moves⟩⟩
```

Subset sum

- In the *n*-queens problem, the goal is a sequence of queen positions, one in each row, such that no two queens attack each other. For each row, the algorithm *decides* where to place the queen.
- In the game tree problem, the goal is a sequence of legal moves, such that each move is as good as possible for the player making it. For each game state, the algorithm *decides* the best possible next move.
- In the SubsetSum problem, the goal is a sequence of input elements that have a particular sum. For each input element, the algorithm *decides* whether to include it in the output sequence or not.