CS101-Quiz9-Review

Suting Chen

CS101-Quiz9-Review

Key Points

1. Topological Sort
2. Greedy Algorithms

Topological Sort

1. An algorithm for ordering the vertices of a directed acyclic graph (DAG) in a linear ordering.
2. Time complexity: $O(V + E)$
3. Space complexity: $O(V)$

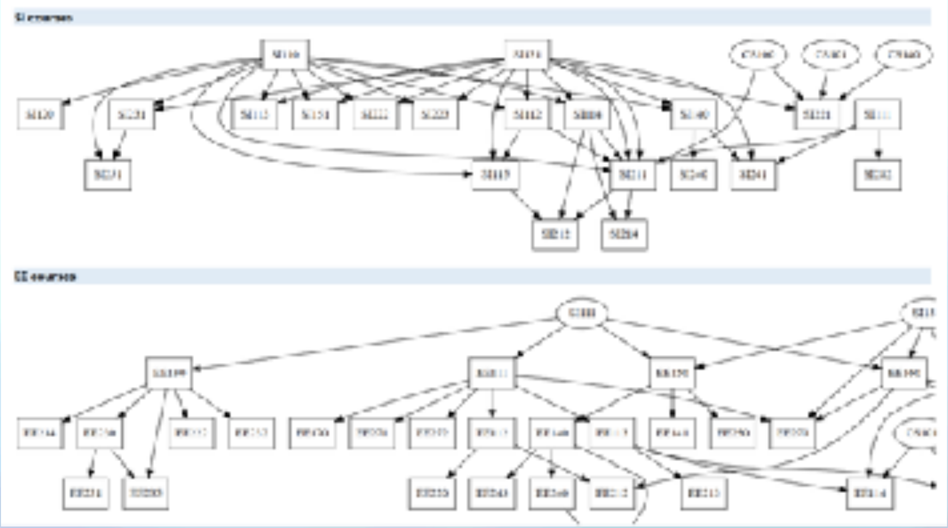
Topological Sort

Basic Theorems and Lemmas

1. A graph is a DAG if and only if it has a topological sorting
2. A DAG always has at least one vertex with in-degree zero.
3. Any sub-graph of a DAG is a DAG.

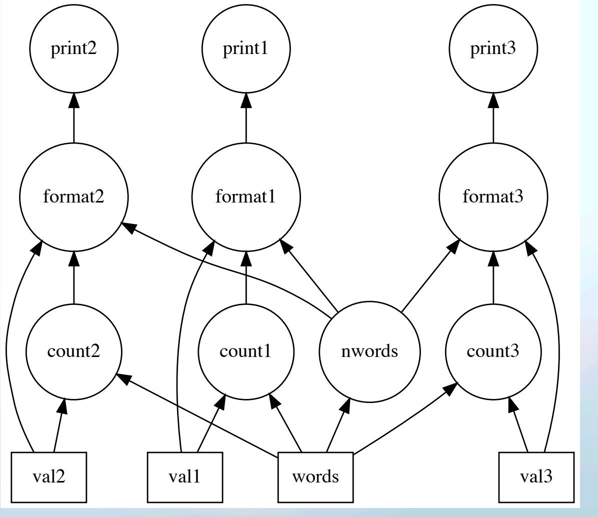
Topological Sort

Curriculum scheduling



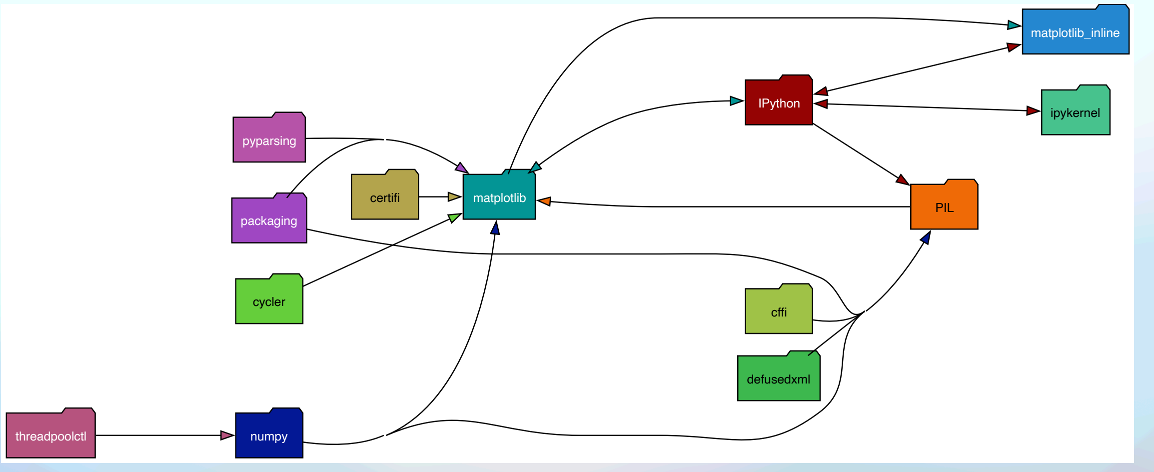
Topological Sort

Task graph — Python Dask



Topological Sort

Task graph — Dependency



CS101-Quiz9-Review

Key Points

1. Topological Sort
2. **Greedy Algorithms**

Greedy Algorithms

Array Section

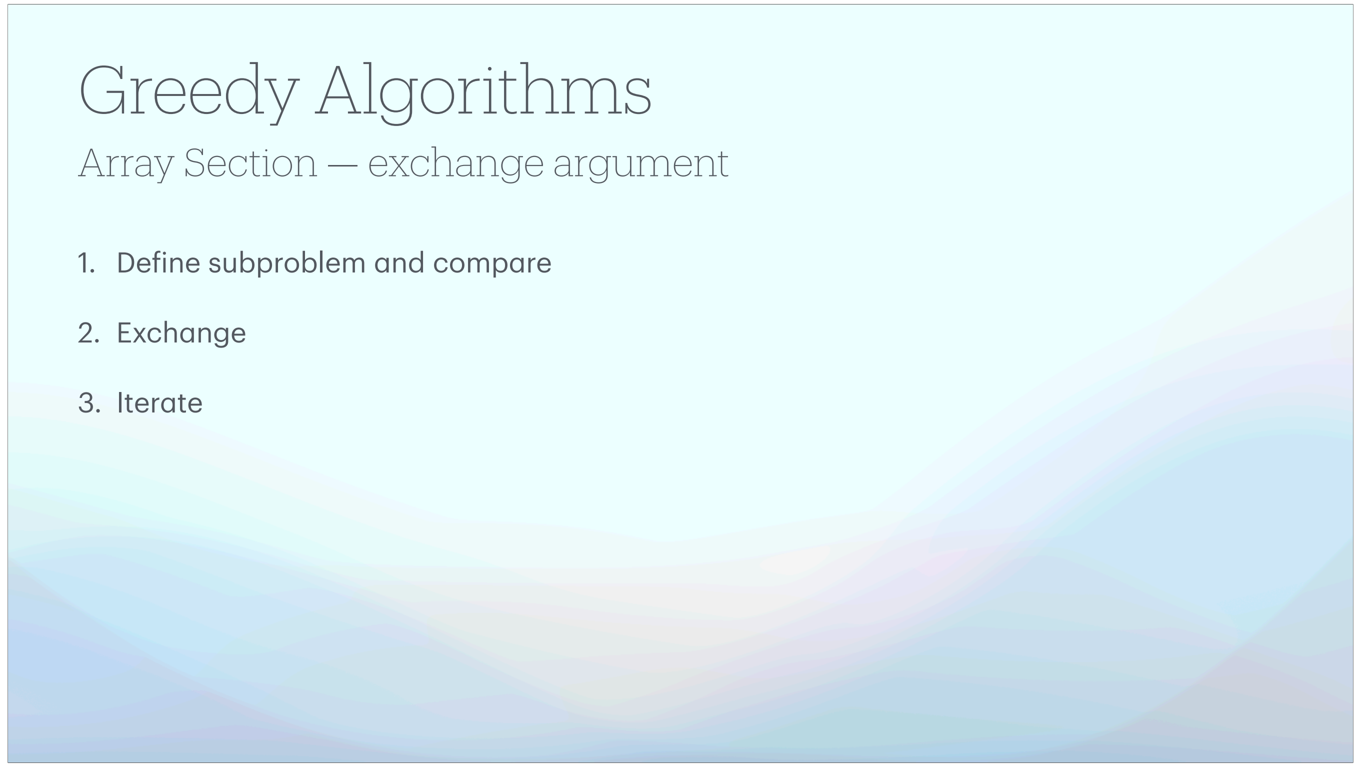
```
def ArraySection(A):
  s = A[1]
  g = 1
  for i = [2, ..., n]:
    s = s + A[i]
    if s > M:
      s = A[i]
      g = g + 1
  return g
```

4	2	4	5	1	2

Greedy Algorithms

Array Section — exchange argument

1. When all optimal solutions have the **same size and differ only in their cost.**
2. Justify with how you will replace some parts of the optimal solution.



这里的compare说的是“我们在贪的东西”和“题面”要求的问题等价。
在部分问题里，这个东西可能非常难证；但是本题不存在这个问题。

Greedy Algorithms

Array Section — exchange argument

1. Sub-problem $g(i)$ means minimizing the number of sections for $\langle a_1, \cdots, a_i \rangle$, and then maximizing the start indices of all the sections.
2. Exchange
3. Iterate

我们的“子问题”的最大规模刚好等价于题面。

Greedy Algorithms

Array Section — exchange argument

1. Sub-problem $g(i)$ means minimizing the number of sections for $\langle a_1, \cdots, a_i \rangle$, and then maximizing the start indices of all the sections.

2. Exchange

3. Iterate

正常来讲这两步要分开，但是因为这个题iteration很显然，所以直接一个数学归纳做完。

Greedy Algorithms

Array Section — exchange argument

1. Sub-problem $g(i)$ means minimizing the number of sections for $\langle a_1, \cdots, a_i \rangle$, and then maximizing the start indices of all the sections.

2. Assume the start index of sections is $\langle s_1, \cdots, s_k \rangle$. Assume we can make s_t smaller.

   - Prove the new solution is not better than our solution (see (e.) in the answer book).

3. Iterate

我知道你想看具体怎么做，但是没空。不如看答案。
这里的重点在于要讲清楚，虽然对于大部分的贪心问题来讲，我们贪心出来的结果都不是唯一的正确答案， 但是只要没有任何一种答案会比他好就可以。

Greedy Algorithms

Array Section — exchange argument

1. Sub-problem $g(i)$ means minimizing the number of sections for $\langle a_1, \cdots, a_i \rangle$, and then maximizing the start indices of all the sections.

2. Assume the start index of sections is $\langle s_1, \cdots, s_k \rangle$. Assume we can make s_t smaller.

   - Prove the new solution is not better than our solution (see (e.) in the answer book).

3. 显然成立

Greedy Algorithms

Array Section — exchange argument — Take-home message

1. The **exchange method** assumes that the "size" of the solution is the same for both greedy and optimal solutions.
2. Exchange is then used to show that it is optimal.
3. Mathematical induction is often used to show how this consistency holds as the problem size increases.

一般来讲，exchange经常会使用的场景下，第三条递推都比较显然。
某种意义上来讲，piazza的pdf说的两种方法其实差不多，其中exchange是一种比较常见的特例。

这里可以保证的是，我们在考试出的greedy证明难度一定是小于作业最后一题的；并且存在一些greedy算法可以这两种方法都不用，就能解决。

如果贪心贪出的结果不仅是正确的，而且还是唯一的；那么我们只需要证明他的唯一性，就可以说明正确性。在这个过程中，唯一性结论本身更强，但是往往更好证明。一会Quiz的题就是这样。