**算法设计与分析 实验报告**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 王少博 | | 院系 | | 软件学院 | | | | 学号 | | 181110315 |
| 任课教师 | | 王金宝 | | | | | 指导教师 | | 王金宝 | | |
| 实验地点 | | 研究院1号楼5？7 | | | | | 实验时间 | | 2020年 月 10 日24 | | |
| 实验名称 | | 贪心算法设计与分析 | | | | | | | | | |
| 同 组 人 | | 无 | | | | | | | | | |
| 实验内容（问题，思路，程序，结果） | | | | | | | | | | | |
| **问题**  设*X*是实数轴上*n*个区间的集合，*X*={[*l*1, *u*1], [*l*2, *u*2], …, [*l1*, *un*]}。假设集合*X*中的所有区间集合覆盖实数轴上连续的区域。*Y*⊆*X*是*X*的子集，如果*Y*中的区间覆盖*X*中的所有区间，即*X*中任意区间内的实数均属于*Y*的某个区间内，则称*Y*为*X*平铺路径。平铺路径的大小是指*Y*中的区间个数。试设计一个贪心算法求解集合*X*中最小的平铺路径。  （**要求：**在算法设计和分析部分说明如下内容：1）贪心策略；2）剩余子问题；3）贪心选择性；4）优化子结构；5）随机生成*X*中的*n*个区间，编写程序实现算法求解*X*的最小的平埔路经）  本作业使用jupyter notebook完成。  **问题分析：**   1. 输入：*X*={[*l*1, *u*1], [*l*2, *u*2], …, [*l1*, *un*]} 2. 输出：*Y=*{[*l*i, *u*i], [*l*i+1, *u*i+1], …, [*lj*, *uj*]}⊆*X*，且*X*中任意区间内的实数均属于*Y*的某个区间内.   **算法：**   1. 排序：   我们对于所有的*l*i,进行排序，如果存在*l*i=*l*j的情况，若有*r*i<*r*j我们把*j*排在前面，否则把*i*排在前面。   1. 贪心策略：   每次选择排序最前面的区间*s*添加到结果集合中，判断*s*区间内是否存在其他区间*t*的左端点*lt*, 如果存在，我们找到*s’ = argmaxt*{*rt | lt∈*[*ls, rs*], *t>s* }。如果这样的区间不存在，或者s’<把*r*s’*<ls*，那么我们找到第一个和区间1不交的集合作为新一轮的第一个区间，否则让*s’*为新一轮的第一个区间。在*s’*之前的所有区间均不予考虑。重复以上操作，顺序遍历完所有的区间。   1. 剩余子问题：   如果第1个(索引为j)区间和某些区间有交集，那么根据贪心策略选择了第2个区间之后，有区间和第二个2区间有交集，交集的部分已经被第2个区间所覆盖，所以子问题的时候不需要考虑交集的区间。因此我们定义子问题如下：  其中*I=*{ *r*[*i*] *| i>j, li∈*[*lj,rj*]}, *s = argmaxi*(*I*)  如果没有和第一个区间有交集的那么剩余子问题就是  下面证明贪心算法的可行性：   1. 贪心选择性：   ***引理1***  设*X*={[*l*1, *u*1], [*l*2, *u*2], …, [*l1*, *un*]} 是实数轴上*n*个区间的集合，且*l*1≤*l*2≤…≤*l*n，且如果有*li=li+1*，我们有*r*i>*r*i+1。那么   1. 区间1 [*l1,r1*]一定在*X*的某个优化解中。 2. 令*I=*{ *r*[*i*] *| i>1, li∈*[*l1,r1*]}, *s = argmaxi*(*I*)。如果*I*非空，那么*s*一定在*X*的某个优化解中。且如果*s*在*X*的某个优化解*A*中*,*那么*I-{s}*集合中的区间均不在*A*中*。*   证明：  首先证明1)  假设*A*为*X*的一个优化解。设*X*的第一个区间是*k*。   * 如果*k=1*，结论成立 * 如果*k>1*，那么我们令*B=A\{k}∪{1}。*设第二个区间为*j*。由于*A*为优化解，所以[*lk, rk*]和[*lj, rj*]必然互相不包含。如果*lk=lj*，那么由于*rk< rj*，矛盾。所以必有*lk<lj。*同时，因为优化解*A*可以覆盖区间1，所以必然可以覆盖*l1*,所以*lk≤l1*，同时*l1≤lk*。所以一定有*l1=lk。*根据排序规则知，*rk≤r1，*所以*l1*可以覆盖*lk。*即*B*也是*X*的一个优化解。   再证明2)  假设*A*为*X*的一个优化解且包含区间*1*。如果*I*非空，那么任意*i∈I*必然有区间*i*和区间*1*有交集。如果*A*中没有*s*，那么必然存在*I*中一个区间*i*在*A*中，否则*I*中区间和1交集的部分无法被覆盖。那么由*s*的定义知，区间{*1}*∪{*s*}可以覆盖所有的*I*中集合，且|B|=|A|所以*B*也是*X*的一个优化解。如果存在*i∈I*, *i∈A*, 且*i≠s*，那么令*B’=|B|-{i}，B’*为*X*的一个优化解。且*|B’|<|B*|，矛盾！  综上证毕。   1. 优化子结构：   ***引理2***  设*X*={[*l*1, *u*1], [*l*2, *u*2], …, [*l1*, *un*]} 是实数轴上*n*个区间的集合，且*l*1≤*l*2≤…≤*l*n，且如果有*li=li+1*，我们有*r*i>*r*i+1。设*A*是*S*的一个优化解，且包含区间1和*s，*其中*I=*{ *r*[*i*] *| i>1, li∈*[*l1,r1*]}, *s = argmaxi*(*I*)。则*A’=A\I*∪{*s*}-{*1*}是剩余子问题  的优化解。  证明：如果*I*为空集，那么区间1和所有区间均不相交，此时*X’=X-{1}*，显然成立。如果*I*不是空集，那么显然，*A’*中的集合保持原先的排序顺序。我们只需要证明*A’*是最小的即可。反之，如果存在*A’’*使得*|A’’|<|A’|*那么我们令*B=A’’*∪*I*∪{*1*}，则由于*A’’*∩*I={s}*，且1不在*A’’*或*I*中，所以|B|=*|A’’|*+|*I|-1+1=|A’’|*+|*I|<|A’|*+|*I|=|A|，*与*A*是优化解矛盾！证毕   1. 算法正确性证明：   ***引理3***  设*X*={[*l*1, *u*1], [*l*2, *u*2], …, [*l1*, *un*]} 是实数轴上*n*个区间的集合，且*l*1≤*l*2≤…≤*l*n，且如果有*li=li+1*，我们有*r*i>*r*i+1。令*Ij=*{ *r*[*i*] *| i>1, li∈*[*lj,rj*]},  *sj =*  *, X1=X。*令*k=min{j|Xj+1=Φ}*，那么*A=s1*∪*s2*∪…∪*sk*为*X*的优化解。  证明：对于*|A|*做归纳法。*|A|=1*时由引理1结论成立。假设对于*|A|≤m-1*结论成立，那么对于*|A|=m,*我们由引理2，不妨令*A*满足*l1=1*。那么我们令|*A1*|=*A-{1}，*由归纳假设知存在优化解*A1=s2*∪*s3*∪…∪*sk*为优化解*，*因此*A=s1*∪*s2*∪…∪*sk。*因此*A*也为优化解，对于*|A|=m*结论成立。   1. 伪代码描述：   initialize *current :=1;*  initialize *result :=φ*;  **while** *current<n*  把*current*区间加入*result*集合;  calculate *I;*  initialize *s;*  initialize *num :=0;*  **if** *I*≠φ  *num :=* I中与current区间有交集的区间个数;  *s := argmaxi(I)*;  **if** *rs<=rcurrent*  *current := current + num + 1;*  **else**  *current := s;*  **endif**  **else**  *current := current + num + 1;*  **endif**  **endwhile** | | | | | | | | | | | |
| 实验结论（结果分析、遇到的困难和解决方法等） | | | | | | | | | 备注 | |  |
| 结果分析如下：  和之前的实验一样，使用可视化直观检验算法是否正确：    横坐标表示区间的序号，在这个例子中我们生成了30个区间。每个区间画一条**垂直线段**，线段**下端点**是区间**左端点**，线段**上端点**是区间**右端点**。红色的线段是我们需要选择的集合的可视化描述，蓝色的线段是我们不需要选择的集合的可视化描述。我们把结果的区间用蓝色的线段首尾相连得到最终的结果集合。  本次实验的困难主要在于证明，其中一个难点就是关于剩余子问题的描述，因为我们每次贪心选择一个区间后需要筛除一些不需要的区间，剩余子问题就不是只删去选择区间的描述（和之前直接去掉排序第一个元素的问题证明略有差异）。另外在证明贪心选择性和优化子结构的时候，具体的数学描述也需要多加考虑和斟酌。比如每次贪心选择的时候可能*I*是空集，这个时候我们就直接选下一个区间即可，不能每次只考虑一种情况。另外就是边界条件，关于等号和不等号的问题，对于算法至关重要，可能因为一个等号写错就得不到优化解了。 | | | | | | | | | | | |
| 教师评价 | | | | 备注 | |  | | 得 分 | |  | |
|  | | | | | | | | | | | |