Problems

Jiankai Sun

October 22, 2014

Question 1. Given n intervals (s_i, f_i) where $1 \leq i \leq n$, s_i and f_i is $i-th$ interval's start and finish time, its duration time is $f_i - s_i$. Use any algorithm to get the maximize the total sum of mutually disjoint intervals.

1. we can use dynamic programming to solve this problem
2. for n intervals (s_i, f_i), it's sorted so that $f_1 \leq f_2 \leq ... \leq f_n$
3. We use $T(i)$ to represent the maximize total sum of intervals of $((s_1, f_1), (s_2, f_2), ...(s_i, f_i))$. each interval has two choices: added to the select set or not.

   Define $p(i)$=the largest$j < i$ that interval i doesn't overlap with j.

   So the recurrence function is: $T(i) = max \begin{cases} T(p(i)) + f_i - s_i \\ T(i-1); otherwise \end{cases}$
4. boundary condition $T(0) = 0$
5. Implement the algorithm

Algorithm 1 function maxSumInterval(i)

Sort the interval according to f_i, so that $f_1 \leq f_2 \leq ... \leq f_n$.
Start time array S=$(0, s_1, s_2, ..., s_n)$, finish time array F=$(0, f_1, f_2, ..., f_n)$.

```
1: global T[0,1,2,...,n], T[0]=0; P[1,...,n]
2:
3: for i ← 1 to n do
4:     if T(p(i))+f_i-s_i¿T(i-1) then
         T[i]=T(p(i))+f_i-s_i
         P[i]=p(i)
5:     else
         T[i]=T[i-1]
         P[i]=-1
6: Return T[n]
```

6. we use $P[1, 2, ..., n]$ to print selected intervals

Algorithm 2 function printInterval()

```
1: global P[1, ..., n]
2: i=n
3: while P[i] ≠ 0 do
4:     if P[i] > 0 then
         print i
         i=P[i]
5: print i
```

7. We use $O(nlog(n))$ to sort. For each i we need $O(n)$ to scan all the interval list and repeat n times, So the time complexity is $O(n^2)$

8. source code: max_selected_intervals.cpp