HW 3

Qn1,

Theorem Optimal substructure of A-S problem:

Let Sij be the set of activities that start after activity ai finishes and that finish before activity aj starts. Let A be the optimal solution set for the set Sij.

Let k be the one activity (anyone) in A.

We define that: Aik = Aij ∩ Sik, which mean Ajk contains the activities in Aij that finish before k starts. And Akj = Aij ∩ Skj, which means Akj contains the activities in Aij that start after k finishes.

1, Aik is the optimal solution for sub-problem Sik

2, Akj is the optimal solution for sub-problem Skj

Proof:

1,

if Ajk is not the optimal solution for sub-problem Sik, then that means we could find a set A’ik of mutually compatible activities in Sik where | A’ik |>| Ajk |, then we could use A’ik, rather than Ajk , in the solution to the sub-problem for Sij, we would have constructed a set of | A’jk |+| Akj |+1 >| Aik |+| Akj |+1= |A| mutually compatible activiries, which contradicts the assumption that A is an optimal solution.

2,

A symmetric argument applies to the activities in Skj

Qn2,

A,

Let X = {xi, xi+1, . . . , xj}, we sort and re-label the points so that

xi ≤ xi+1 ≤ · · · ≤ xj.

Let Sij be the smallest set of intervals we build (the solution), contains some intervals Ik = [xk , xk + 1], where i ≤ k ≤ j.

Same definition for Sjk and Sjk.

Xik = Sjk  ∩ X

Xkj = Sjk  ∩ X

1, Sik is the optimal solution for sub-problem Xik

2, Skj is the optimal solution for sub-problem Xkj

B,

Consider any nonempty sub-problem Xk, and let x1 be a point in Xk with the smallest position. Then interval I1 = [x1, x1+1] is included in some of smallest set of intervals of Xk.

C,

Let Sk be the smallest set of unit-length closed intervals of Xk, and let I’1 = [x’1,x’1+1]be the first interval in Sk. If I’1 =I1, then we are done, since we have shown that I1 is in some smallest set of intervals of Xk.

I’1=[x’1,x’1+1], and I1 = [x1, x1+1].

If x’1 != x1:

If x’1 > x1, we can say Sk is not a solution for the question since the first point x1 isn’t included in Sk.

If x’1 < x1, as x1 is the leftmost point, there are no points from Xk contained in the interval [x’1,x1]. Therefore, we could simply replace the interval [x’1,x’1+1] in Sk ( which is Sopt) with the interval [x1, x1+1] such that the new set of intervals Sk-revise is still optimal (as | Sk | = | Sk\_revise | and all points are covered).

D,

Recursive:

s= set of intervals (solution)

X = set of the given points

initial s = Φ

Unit-length-closed-intervals\_ Recursive (s, X):

1, if X = Φ

2, return s

3,else:

4, sort(X)

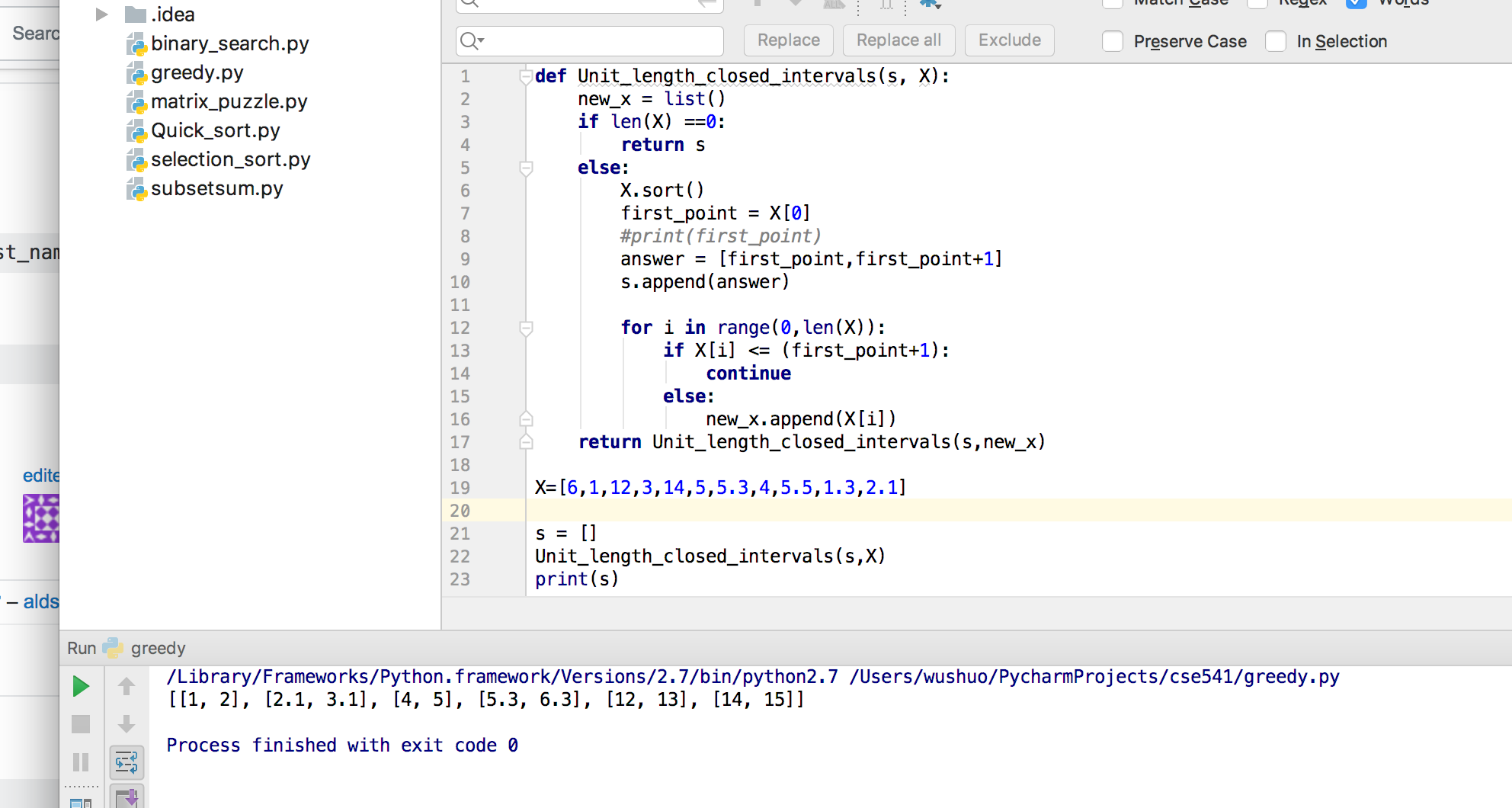
5, first\_point <- X[0]

6, Ik = [first\_point, first\_point +1]

7, s <- (s U {Ik})

8, new\_X <- (X - {points: points ∈Ik}

9, return Unit-length-closed-intervals(s,new\_X)



E,

Iterative:

s= set of intervals (solution)

X = set of the given points

initial s = Φ

Unit-length-closed-intervals\_ Iterative(s, X):

1, sort(X)

2, while X != Φ:

3, for each point pt in X:

4, first\_point <- X[0]

5, Ik = [first\_point, first\_point +1]

6, s = (s U {Ik})

7, X = (X - {points: points ∈Ik}

8, return s

