Dynamic Programming | Set 20 (Maximum Length Chain of Pairs)

You are given n pairs of numbers. In every pair, the first number is always smaller than the second number. A pair (c, d) can follow another pair (a, b) if b < c. Chain of pairs can be formed in this fashion. Find the longest chain which can be formed from a given set of pairs.
Source: Amazon Interview | Set 2

For example, if the given pairs are {{5, 24}, {39, 60}, {15, 28}, {27, 40}, {50, 90} }, then the longest chain that can be formed is of length 3, and the chain is {{5, 24}, {27, 40}, {50, 90}}

This problem is a variation of standard Longest Increasing Subsequence problem. Following is a simple two step process.

1) Sort given pairs in increasing order of first (or smaller) element.
2) Now run a modified LIS process where we compare the second element of already finalized LIS with the first element of new LIS being constructed.

The following code is a slight modification of method 2 of this post.

```
#include<stdio.h>
#include<stdlib.h>

// Structure for a pair
struct pair
{
  int a;
  int b;
};

// This function assumes that arr[] is sorted in increasing order
// according the first (or smaller) values in pairs.
int maxChainLength( struct pair arr[], int n)
{
   int i, j, max = 0;
   int *mcl = (int*) malloc ( sizeof( int ) * n );

   /* Initialize MCL (max chain length) values for all indexes */
   for ( i = 0; i < n; i++ )
      mcl[i] = 1;

   /* Compute optimized chain length values in bottom up manner */
   for ( i = 1; i < n; i++ )
      for ( j = 0; j < i; j++ )
         if ( arr[i].a > arr[j].b && mcl[i] < mcl[j] + 1)
            mcl[i] = mcl[j] + 1;

   // mcl[i] now stores the maximum chain length ending with pair i

   /* Pick maximum of all MCL values */
   for ( i = 0; i < n; i++ )
      if ( max < mcl[i] )
         max = mcl[i];

   /* Free memory to avoid memory leak */
   free( mcl );

   return max;
}


/* Driver program to test above function */
int main()
{
    struct pair arr[] = { {5, 24}, {15, 25},
                          {27, 40}, {50, 60} };
    int n = sizeof(arr)/sizeof(arr[0]);
```

```
    printf("Length of maximum size chain is %d\n",
           maxChainLength( arr, n ));
    return 0;
}
```

Output:

```
Length of maximum size chain is 3
```

Time Complexity: O(n^2) where n is the number of pairs.

The given problem is also a variation of Activity Selection problem and can be solved in (nLogn) time. To solve it as a activity selection problem, consider the first element of a pair as start time in activity selection problem, and the second element of pair as end time. Thanks to Palash for suggesting this approach.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Topics:

- Find Union and Intersection of two unsorted arrays
- Pythagorean Triplet in an array
- Maximum profit by buying and selling a share at most twice
- Design a data structure that supports insert, delete, search and getRandom in constant time
- Print missing elements that lie in range 0 – 99
- Iterative Merge Sort
- Group multiple occurrence of array elements ordered by first occurrence
- Given a sorted and rotated array, find if there is a pair with a given sum

Tags: Dynamic Programming

Writing code in comment? Please use **ideone.com** and share the link here.

Here if we are sorting the pairs by first element than how are we making sure that elements will be choosed from the original given order
I think sorting must be banned here

• Reply • Share ›

Siren • 7 months ago

If you have learned Longest increasing subsequence, then this is exactly like that ...just have to change the comparision operator with a function..

```
class pair{

public int a;

public int b;

pair(int a, int b){

this.a = a;

this.b = b;

}

public int getA() {

return this.a;
```

see more

• Reply • Share ›

GuojiaAgain • 7 months ago

Classic Greedy Problem, sort with second value, and always choose possible interval with lowest second value.

4 • Reply • Share ›

richa • 8 months ago

it seems that using greedy is better but even if it has to be a dp solution then too why can't we sort acc to 2nd element ??

• Reply • Share ›

nithin • 8 months ago

@GeeksforGeeks : isn't this a better soln?
1.sort arr[] according to arr[i].a).
2.declare count array of n size and count[0]=1.
3.for i=1 to n-1
do

```
do
j=i-1;
while ( j>0 && arr[j].b >= arr[i].a)
j--;
if (j==0)
count[i]=1;
else
count[i]=count[j]+1;
end for
```

4. return max in count[]

Shashwat • 8 months ago

I think this is the famous task scheduling. It has a greedy solution.
Sort the pairs in increasing order of second number.
Always chose the one with the minimum second number.

This problem would have been a DP (similar to LIS) if the pairs had to be in sequence.
L[i] = length of such sequence for a[0...i];
L[0] = 1
L[i] = 1 + max(L[j]) for all j<i, only="" if="" a[i]="">> b[j]... right?

Anshum agrawal • 10 months ago

why do we have to sort the array in Step 1 ?

AlienOnEarth • a year ago

Algorithm for Greedy Approach:

1.) Sort the array based on last element (Finish Time)
2.) Apply O(o^2) Algorithm to find all the pairs which satisfy Pairing property. While doing this, keep track of max pair count
3.) return max pair count

CodeGame ➦ AlienOnEarth • a year ago

no need of any 0(o^2). Just do a linear scan on sorted result u get the result. Idea is to choose the one ending first and not conflicting with previous selection.

smith • a year ago

what is need of sorting first

Vivek ➚ smith • a year ago

because here the original sequence is of no importance.

In conventional lis problem , we have to follow the original sequence order but here no such limitation is there

• Reply • Share ›

danny ➚ Vivek • 10 months ago

But without sorting also, LIS can be found in this case also then why to have an overhead.....

• Reply • Share ›

Anurag Singh ➚ danny • 9 months ago

Both numbers of pair should be increasing and it should be maximum possible pair. If you don't sort, you will get pairs in increasing order, but it won't be maximum possible pairs.
But anyway, best way to solve this problem is Greedy way. This is nothing but activity selection problem as it is explained by GeeksforGeeks at the end above.

2 • Reply • Share ›

Shiwakant Bharti • 2 years ago

Hi Omar/All,
I have attached Java solution for the same. Please have a look and comment.

1 • Reply • Share ›

shiwakant.bharti • 2 years ago

Here is Java adaptation as requested by Omar.
Note: This is just contents which can be included inside any class.

```
static class Pair implements Comparable<Pair> {
   /**
    * @param a
    * @param b
    */
   public Pair(int a, int b) {
       super();
       this.a = a;
       this.b = b;
   }

   public int a, b;

   /*
```

```
 * (non-Javadoc)
```

see more

• Reply • Share ›

shiwakant.bharti ➚ shiwakant.bharti • 2 years ago

Admin: Looks like there is some formatting problem here, kindly copy it to eclipse to see the proper code and comments. I used the language="Java" do I need type in "java" or "JAVA" instead?

• Reply • Share ›

Krum Bakalsky • 2 years ago

I think that this is simply solved in O(n) time, because it is an instance of the longest-path-in-DAG problem, which can be solved in O(V + E) time.

• Reply • Share ›

suri ➚ Krum Bakalsky • 8 months ago

But you can't form a DAG out of intervals in linear time.

• Reply • Share ›

Silent • 2 years ago

i guess we can simply do it using greedy approach by sorting int the increasing order of second elements.. plz reply fast??

• Reply • Share ›

suri ➚ Silent • 8 months ago

But you can't form a DAG out of intervals in linear time.

• Reply • Share ›

gargsanjay • 2 years ago

after sorting....
step 1) pick the pair with smallest b value.
step 2)among the pairs with a value greater than b value of previous node in mcl, choose the one with least b value.
what abt this algorithm..

• Reply • Share ›

Omar Hidayat • 2 years ago

how to convert in code pair to java.

• Reply • Share ›

abhishek08aug • 2 years ago

Intelligent :D

```
/* Paste your code here (You may delete these lines if not writing code) */
```

• Reply • Share ›

Napster • 2 years ago

Dont you think ,with activity selection algo,order of pairs would be depleted.

```
/* Paste your code here (You may delete these lines if not writing code) */
```

• Reply • Share ›

Arun • 2 years ago

I believe that the above question can be easily solved using this approach:

1. Sort the pairs based in the increasing order of second values, e.g. (5,24), (15, 28), (27, 40), (39, 60) and (50, 90)

2. Now add the first pair to the list.

3. Start with i = 0; and check if y(i) < x(i+1), then add x(i+1)(i+1) to the list, else discard it and move on to next pair.

4. Continue iterating over the list till i == size of the given pairs....

Done !!!

1 • Reply • Share ›

makeit ➚ Arun • 2 years ago

What if the given pairs are (1, 100),5,24), (15, 28), (27, 40), (39, 60) and (50, 90)?

1.sort the list,
2. add the first one first.
3. then the result is 1,
4. not good

or else:
for each pair in the list, consider it as the first pair in the list.
In this case, the overall run time is still o(n**2)

```
/* Paste your code here (You may delete these lines if not writing code) */
```

• Reply • Share ›

hardhik ➚ makeit • 5 months ago

He was asking you to sort acc. to second values !

1 • Reply • Share ›

yesminister Arun • 2 years ago

Arun: You are right. In fact that's exactly the method employed in the activity selection algorithm linked in the explanation above.

• Reply • Share ›

VikasG Arun • 2 years ago

I think you have misunderstood the problem. You are not allowed to change the order of the pairs in the input. By sorting, you are solving the problem for a different input.

• Reply • Share ›

rahul • 3 years ago

Why sorting??

```
/* Paste your code here (You may delete these lines if not writing code) */
```

2 • Reply • Share ›

zoe rahul • 2 years ago

Yeah can't understand why we need sorting here...
Can be done using LIS method exactly without sorting...

```
/* Paste your code here (You may delete these lines if not writing code) */
```

• Reply • Share ›

its_dark zoe • 7 months ago

You will get some sequence w/o using sorting. but it won't be guaranteed that it would be of maximum length.

• Reply • Share ›

Zavatar zoe • 2 years ago

It confused me too at first.

But then I realized the result is constructed from a given set of pairs, not a subsequence.

1 • Reply • Share ›

Palash • 3 years ago

Isn't this the standard job scheduling problem and can be done in NlogN by greedy approach, by selecting the one with nearest end time at each step ?

1 • Reply • Share ›

Shashwat Palash • 8 months ago

Yes, I too thought so.
Always chose the one with the smallest second value.

• Reply • Share ›

Abhinav Palash • 2 years ago

Exactly!
Thats what I thought!

```
/* Paste your code here (You may delete these lines if not writing code) */
```

• Reply • Share ›

kartik Palash • 3 years ago

@Paiash: Thanks for sharing your thoughts. This in fact looks similar to activity selection problem

• Reply • Share ›

Palash • 3 years ago

Isn't this, job scheduling problem, and can be done in nlogn, by greedy ?

```
/* Paste your code here (You may delete these lines if not writing code) */
```

• Reply • Share ›

anonymous • 3 years ago

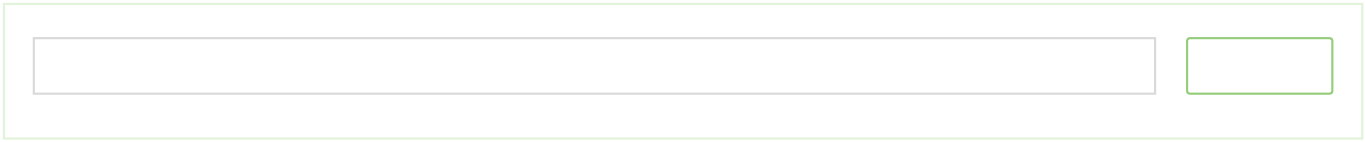
is it not simple?

just sort them based on the second values

pick the next one which has first value greater than the current one's second value

```
/* Paste your code here (You may delete these lines if not writing code) */
```

• Reply • Share ›



- Interview Experiences
- Advanced Data Structures
- Dynamic Programming
- Greedy Algorithms
- Backtracking
- Pattern Searching
- Divide & Conquer
- Mathematical Algorithms
- Recursion
- Geometric Algorithms

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

Follow @GeeksforGeeks

Recent Comments

- lt_k

  i need help for coding this function in java...

Java Programming Language · 1 hour ago

- Piyush

  What is the purpose of else if (recStack[*i])...

  Detect Cycle in a Directed Graph · 1 hour ago

- Andy Toh

  My compile-time solution, which agrees with the...

  Dynamic Programming | Set 16 (Floyd Warshall Algorithm) · 1 hour ago

- lucy

  because we first fill zero in first col and...

  Dynamic Programming | Set 29 (Longest Common Substring) · 2 hours ago

- lucy

  @GeeksforGeeks i don't n know what is this long...

  Dynamic Programming | Set 28 (Minimum insertions to form a palindrome) · 3 hours ago

- manish

  Because TAN is not a subsequence of RANT. ANT...

  Given two strings, find if first string is a subsequence of second · 3 hours ago