Divide and Rule Dynamic Programming (DP) Algorithms Computer Programming

What is divide and conquer optimization in dynamic programming?

http://codeforces.com/blog/entry/8219

This question previously had details. They are now in a comment.

Answer Follow · 17 Request 1

2 Answers

Michael Levin, I teach Algorithms: https://goo.gl/MROslP and https://goo.gl/F1vtbF
Updated Apr 5

This is an optimization for computing the values of Dynamic Programming (DP) of the form $dp[i][j] = \min\limits_{k<j}(dp[i-1][k] + C[k+1][j])$ for some arbitrary cost function $C[i][j]$ such that the following property can be proved about this dynamic programming with this cost function. Let's denote by $A[i][j]$ the optimal k for which $dp[i][j] = dp[i-1][k] + C[k][j]$. The property is that for any i and j, $A[i][j] \le A[i][j+1]$, that is, the optimal k is monotone on j for fixed i.

An example of such DP is the following problem: given n objects with weights $w_1, w_2, \ldots, w_n$, divide them into m groups of ***consecutive*** objects, such that the sum of squares of total weights of the groups is minimal (total weight of the group is the sum of weights of the objects in the group). It can be proved that the optimal k in this problem is monotone on j. The same can be proved if we take cost function $W \log W$ for the total weight W of the group given that all the weights are positive (minimize the sum of $W \log W$ among all distributions of n objects into m groups) or any other ***convex function*** of the total weight of the group. In these cases, $C[i][j]$ in the DP formulation would be the cost function for a group with objects from i to j inclusive.

The straighforward solution of this DP is $O(mn^2)$, because we need a loop over i (m iterations), a loop over j (n iterations) and a loop over k for each j ($n/2$ iterations on average). However, given the monotonicity condition $A[i][j] \le A[i][j+1]$, this DP can be solved in $O(mn \log n)$. More specifically, for each fixed i we will solve the iteration in $O(n \log n)$ instead of $O(n^2)$.

This can be done by the following pseudocode which for fixed i applies divide and conquer on j and keeps range $(jleft, jright)$ of values of j for which we are seeking the answer and the corresponding range $(kleft, kright)$ of the possible values for optimal k when j is in the range $(jleft, jright)$:

```
def ComputeDP(i, jleft, jright, kleft, kright):
  # Select the middle point
  jmid = (jleft + jright) / 2
```

There's more on Quora...

Pick new people and topics to follow and see the best answers on Quora.

Update Your Interests

Related Questions

What is Knuth's optimization in dynamic programming?

What is the difference between dynamic programming and divide and conquer?

In an intuitive explanation what is the difference between Divide & Conquer (D & C) and Dynamic Programming (DP)?

What are the most unknown dynamic programming optimization techniques?

How important is dynamic programming for programming interviews?

Why is dynamic programming called dynamic programming?

What are differences between dynamic, divide-and-conquer, and greedy programming?

What are some tricks to optimize dynamic programming solutions?

Why does divide and conquer not have an optimal substructure?

How do we determine whether to use divide & conquer or dynamic programming for certain class of problems?

+ Ask New Question

More Related Questions

```
    bestk = -1
    for k in range(kleft, jmid):
      if dp[i - 1][k] + C[k + 1][jmid] < best:
        dp[i][jmid] = dp[i - 1][k] + C[k + 1][jmid]
        bestk = k
    # Divide and conquer
    if jleft < jmid - 1:
      ComputeDP(i, jleft, jmid - 1, kleft, bestk)
    if jleft + 1 < jright:
      ComputeDP(i, jmid + 1, jright, bestk, kright)

def ComputeFullDP:
    Initialize dp for i = 0 somehow
    for i in range(1, m):
      ComputeDP(i, 0, n, 0, n)
```

Turns out that `ComputeDP` works for $O(n \log n)$ for any fixed i. Actually, a `ComputeDP(i, jleft, jright, kleft, kright)` call works in time $O(q \log p)$ where $p = jright - jleft$ is the length of the range for j and $q = kright - kleft$ is the length of the range for k. The recurrence relation here is

$$T(p, q) = O(q) + T(\tfrac{p}{2}, a) + T(\tfrac{p}{2}, q - a) < Bq + T(\tfrac{p}{2}, a) + T(\tfrac{p}{2}, q - a)$$

for some constant B. By computing it further, we see

$$T(p, q) < Bq + Ba + B(q - a) + T(\tfrac{p}{4}, a') + T(\tfrac{p}{4}, a - a') + T(\tfrac{p}{4}, q') + T(\tfrac{p}{4}, q - a - q') =$$
$$= 2Bq + T(\tfrac{p}{4}, a') + T(\tfrac{p}{4}, a - a') + T(\tfrac{p}{4}, q') + T(\tfrac{p}{4}, q - a - q') < \ldots <$$
$$< \log_2(p) Bq = O(q \log p).$$

7.7k Views · View Upvoters · Answer requested by Saad Taame

Upvote · 81 Share

Add a comment... Comment Recommended All

Claudio Brandolino, Computery guy.
Answered Jun 25, 2016

Prelude: In a group of 100 people, find the two with the smallest square of age difference.

```
personOneBirthday = -1009065600;
// birthdays are expressed in unix timestamps set at the start of the da
personTwoBirthday = -722563200;
ageDifferenceSquare = (personOneBirthday - (personTwoBirthday))^2;
```

We are interested in finding the couple of people with the lowest $ageDifferenceSquare$ among them.

The naive solution would be calculating it for each possible pair. We will get to our result, eventually, but it would take quadratic time (that is, for n points, it would take n^2 calculations).

To speed up the process, we divide the group between Knuth and Sussman fans. We only compare the birth months of each group's members with others in that group, and get the closest pair for each.

At the end, we compare the two groups' winning pairs (and check if there's no closest pair crossing the halves[1]).

Now the comparisons are just $2(n/2)^2$, which is good.

If we have two cores, we can let each core take care of a group, which is better: we only have to wait for $(n/2)^2$ comparisons!

Wanna divide them further amongst Vim and Emacs supporters?

Just $4(n/4)^2$ comparisons!

More cores? $(n/4)^2$

> Guys? Where are you guys? Should I just wait here while you're done?
> - *Core in charge of the Sussman/Vim quadrant*

Incalzando: What's dynamic programming?

It's a technique we use when we have a large set of similar problems where we suspect there will be similar result: we can store each result in a data structure, so we won't have to recalculate it.

For instance, we can gradually fill up an mxm matrix where m is the number of available birthdays: the first row and column will contain a vector v, filled with all of the birthdays in our space.

At each comparison we'd check our matrix to see if the $ageDifferenceSquare$ is already there: at $m(personOneBirthday, personTwoBirthday)$, otherwise we swap the cell with our result.

Scherzo: Was it all wrong then? Exercises for the reader.

- Is a matrix the best data type? Won't some calculations be repeated anyway?
- Is our example's dataset worth the effort? How many people are likely to share a birthday in our dataset? What if we didn't care about the year?
- In which group would I end up? Initially I've been bold and spelled it out, which is completely backwards.
- This was a huge waste of time - complexity wise - time wise. Would it have been the same if we were looking for the two people with the closest sum of $ageDifferenceSquare$ between each one and one fixed date?

Footnotes

[1] A Divide-and-Conquer Approach

Home 1 Answer Spaces Notifications 3 Add Question

Upvote · 3 Share

Add a comment... Comment Recommended All

Rafe Zayed wrote this · Tue

Which is better: CSE at IUT with scholarship or ME/CE at BUET?

Rafe Zayed, studies BSc Mechanical Engineering at Bangladesh University of Engineering and Technology (2019)

Answered Tue

Every year after the admission result i get same type of question with similar error. Subject should not be chosen based on university, rather it should be chosen based on your interest.

Read In Feed

Muntasir Wahed wrote this · Thu

What are, in your opinion, the most important parts of Bengali culture?

Muntasir Wahed, Lived in Bangladesh since 1995, except for 15 days in India

Answered Jun 2, 2017

Ah, Bengalis! The Bengalis (*Bangali*) are a major Indo-Aryan ethno-linguistic group. They are native to the region of Bengal in South Asia, which is

Read In Feed

Tanzir Islam upvoted this · 2m

What is something that a lot of people yearn for, but is often a curse for the ones who have it?

Liz Kieran, Living w/Lupus SLE & much more. Mother. (1989-present)

Answered Nov 4

BIG BOOBS. Yeah, they look awesome in pictures (Ariel Winter looks beautiful here) and outfits look amazing and can get the attention of most guys

Read In Feed