indy256's blog

Dynamic Programming Optimizations

By **indy256**, 3 years ago,

Several recent problems on Codeforces concerned dynamic programming optimization techniques.

The following table summarizes methods known to me.

Name	Original Recurrence	Sufficient Condition of Applicability	O C
Convex Hull Optimization1	$dp[i] = min_{j<i}\{dp[j] + b[j] \star a[i]\}$	$b[j] \geq b[j+1]$ ~~optionally~~ $a[i] \leq a[i+1]$	C
Convex Hull Optimization2	$dp[i][j] = min_{k<j}\{dp[i-1][k] + b[k] * a[j]\}$	$b[k] \geq b[k+1]$ ~~optionally~~ $a[j] \leq a[j+1]$	C
Divide and Conquer Optimization	$dp[i][j] = min_{k<j}\{dp[i-1][k] + C[k][j]\}$	$A[i][j] \leq A[i][j+1]$	C
Knuth Optimization	$dp[i][j] = min_{i<k<j}\{dp[i][k] + dp[k][j]\} + C[i][j]$	$A[i,j-1] \leq A[i,j] \leq A[i+1,j]$	C

Notes:

- $A[i][j]$ — the smallest k that gives optimal answer, for example in $dp[i][j] = dp[i-1][k] + C[k][j]$
- $C[i][j]$ — some given cost function
- We can generalize a bit in the following way: $dp[i] = min_{j<i}\{F[j] + b[j] * a[i]\}$, where $F[j]$ is computed from $dp[j]$ in constant time.
- It looks like **Convex Hull Optimization2** is a special case of **Divide and Conquer Optimization**.
- It is claimed (in the references) that **Knuth Optimization** is applicable if $C[i][j]$ satisfies the following 2 conditions:
- **quadrangle inequality**: $C[a][c] + C[b][d] \leq C[a][d] + C[b][c],\ a \leq b \leq c \leq d$
- **monotonicity**: $C[b][c] \leq C[a][d],\ a \leq b \leq c \leq d$
- It is claimed (in the references) that the recurrence $dp[j] = min_{i<j}\{dp[i] + C[i][j]\}$ can be solved in $O(nlogn)$ (and even $O(n)$) if $C[i][j]$ satisfies **quadrangle inequality**. **WJMZBMR** described how to solve some case of this problem.

Open questions:

1. Are there any other optimization techniques?
2. What is the sufficient condition of applying **Divide and Conquer Optimization** in terms of function $C[i][j]$? Answered

References:

- *"Efficient dynamic programming using quadrangle inequalities" by F. Frances Yao.* find
- *"Speed-Up in Dynamic Programming" by F. Frances Yao.* find
- *"The Least Weight Subsequence Problem" by D. S. Hirschberg, L. L. Larmore.* find

- *"Dynamic programming with convexity, concavity and sparsity" by Zvi Galil, Kunsoo Park.* find
- *"A Linear-Time Algorithm for Concave One-Dimensional Dynamic Programming" by Zvi Galil, Kunsoo Park.* find

Please, share your knowledge and links on the topic.

dynamic programming, **knuth optimization**, **convex hull optimization**

+388 indy256 3 years ago 44

Comments (44)

Write comment?


WJMZBMR

3 years ago, # | ← Rev. 4 +27

Here is another way to optimize some 1D1D dynamic programming problem that I know.

Suppose that the old choice will only be worse compare to the new choice(it is quite common in such kind of problems).

Then suppose at current time we are deal with dp_i, and we have some choice $a_0 < a_1 < a_2, ..., a_{k-1} < a_k$. then we know at current time a_i should be better than a_{i+1}. Otherwise it will never be better than a_{i+1},so it is useless.

we can use a deque to store all the a_i.

And Also Let us denote $D(a, b)$ as the smallest i such that choice b will be better than a.

If $D(a_i, a_{i+1}) > D(a_{i+1}, a_{i+2})$,we can find a_{i+1} is also useless because when it overpass a_i,it is already overpass by a_{i+2}.

So we also let $D(a_i, a_{i+1}) < D(a_{i+1}, a_{i+2})$. then we can find the overpass will only happen at the front of the deque.

So we can maintain this deque quickly, and if we can solve $D(a, b)$ in $O(1)$,it can run in $O(n)$.

→ Reply


kingofnumbers

3 years ago, # ^ | +3

could you please give some example problems?

→ Reply

cgy4ever

3 years ago, # | +5

For question 2: The sufficient condition is:
$C[a][d] + C[b][c] \geq C[a][c] + C[b][d]$ where $a < b < c < d$.

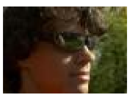
→ Reply


wanbo

3 years ago, # ^ | 0

`Is it quadrangle inequalities? ∀i≤ j,w[i, j]+w[i+1, j+1]≤w[i+1, j]+w[i, j+1], and are these two inequalities equivalent except the >= & <=?`

→ Reply


Sammarize

3 years ago, # | +18

There is one more optimization of dimanic progamming: 101E - Candies and Stones (editoral)

→ Reply

new, 11 days ago, # ^ | +3

More Problem Collection


khatribiru

More Problem Collection.
→ Reply


kingofnumbers

3 years ago, # | ☆ +13

you have put problem "B. Cats Transport" in "Convex Hull Optimization1", actually it belongs to "Convex Hull Optimization2"
→ Reply


indy256

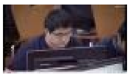
3 years ago, # ^ | ☆ +5

fixed
→ Reply


Zlobober

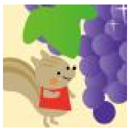
3 years ago, # | ☆ ← Rev. 2 +55

For this moment it's the most useful topic of this year. Exactly in the middle: June 30th, 2013.
→ Reply


MarioYC

3 years ago, # | ☆ +8

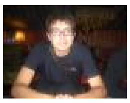
this one seemed a nice dp with optimization to me:https://www.hackerrank.com/contests/monthly/challenges/alien-languages
→ Reply


hogloid

3 years ago, # | ☆ ← Rev. 4 +29

The problem mentioned in the article (Breaking Strings) is "Optimal Binary Search Tree Problem" , traditional one.

It can be solved by simple DP in O(N^3), by using Knuth's optimization , in O(N^2) . But it still can be solved in O(NlogN) — http://poj.org/problem?id=1738 (same problem but bigger testcases) (I don't know how to solve it. I hear the algorithm uses meld-able heap)
→ Reply


Giorgos_Christoglou

3 years ago, # | ☆ +20

Convex Hull Optimization 1 Problems:

- APIO 2010 task Commando
- TRAKA
- ACQUIRE
- SkyScrapers (+Data Structures)

Convex Hull Optimization 2 Problems:

- BAABO

Convex Hull Optimization 3 Problems (No conditions for a[] array and b[] array) :

- GOODG
- BOI 2012 Day 2 Balls
- Cow School
- Solution-Video

→ Reply


victorsenam

10 months ago, # ^ | ☆ 0

GOODG can be solved with Type 1
→ Reply


gninrael

6 months ago, # ^ | ☆ 0

How? I noticed that, in this problem, b[j] follows no order and a[i] can be either decreasing or increasing, depending on how the equation is modeled. I was able to solve it using the fully dynamic variant, but I can't see

solve it using the fully dynamic variant, but I can't see
how to apply the "type 1" optimization.
→ Reply

samier_aldroubi

6 months ago, # ^ | ☆ ← Rev. 2 0

Can you add a link to your code I tried to implement the dynamic variant few weeks ago but there were so many bugs in my code :(.Maybe yours can help :/ .
→ Reply


victorsenam

10 months ago, # ^ | ☆ +3

New link for Commando:

http://www.spoj.com/problems/APIO10A/
→ Reply


zscefn

3 years ago, # | ☆ 0

For some reason I cannot open the links with firefox because they go over the Top Rated table.
→ Reply


indy256

3 years ago, # ^ | ☆ +4

Try to zoom out, pressing Ctrl + -
→ Reply


Monyura

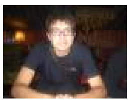
3 years ago, # | ☆ ← Rev. 2 +8

One more problem where Knuth Optimization is used:
Andrew Stankevich Contest 10, Problem C.
BTW, does anybody know how to insert a direct link to a problem from gyms?
→ Reply


mbrc

2 years ago, # | ☆ 0

I need some problems to solve on Divide and Conquer Optimization. Where can I find them? An online judge / testdata available would be helpful.
→ Reply


Giorgos_Christoglou

2 years ago, # ^ | ☆ +1

Check this one : Guardians of the Lunatics
→ Reply


mbrc

2 years ago, # ^ | ☆ 0

Learnt Divide and Conquer Optimization just from there. :P That is why I'm asking for more problems to practice. :D
→ Reply


sifrit98

13 months ago, # ^ | ☆ 0

Is this the best complexity for this problem? Can't we do any better? Can't we somehow turn the logL needed into a constant?
→ Reply

13 months ago, # ^ | ☆ 0

We can, using that `opt[i-1][j] <= opt[i][j] <= opt[i][j+1]` .

Key thing is to see that opt function is monotone for both arguments. With that observation, we

don't need to use binary search

micklepru

don't need to use binary search.

Check out my submission.

→ Reply

2 years ago, # | ☆ +3

92anurag

can anyone provide me good editorial for dp with bitmask .

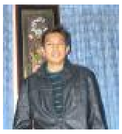
→ Reply

I_love_Meta_MZ

2 years ago, # | ☆ 0

Has matrix-exponent optimizations been included here?

→ Reply



Farsid

2 years ago, # | ☆ +2

Can matrix chain multiplication problem b also optimized by knuth optimization? If not, dn why?

→ Reply

indy256

2 years ago, # ^ | ☆ +3

Quote from the first of the references above:

> *The monotonicity property for the division points does not hold for the matrix multiplication chain problem...*
>
> *Consider the matrices M1,M2,M3,M4 with dimensions 2x3, 3x2, 2x10, and 10x1, respectively. As can be easily verified, the proper order to compute M1M2M3 is to parenthesize it as (M1M2)M3, while the optimal computation of M1M2M3M4 corresponds to M1(M2(M3M4)).*

The second reference gives $O(n^2)$ dynamic programming solution, based on some properties of the matrix chain multiplication problem.

There is also an $O(n * \log n)$ algorithm by Hu and Shing.

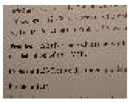
→ Reply

Thomas_Ahle

18 months ago, # ^ | ☆ 0

Link to the Hu and Shing algorithm?

→ Reply



gsCEA

4 months ago, # ^ | ☆ 0

Here is a link to a 1981 version of the thesis. The original was published in two parts in 1982 and 1984.

http://i.stanford.edu/pub/cstr/reports/cs/tr/81/875/CS-TR-81-875.pdf

However, I doubt that this will be used in competitive programming.

→ Reply

mayankp

20 months ago, # | ☆ +1

What are some recent USACO questions that use this technique or variations of it?

→ Reply

16 months ago, # | ☆ ← Rev. 6 0

Can this problem be solved using convex hull optimization?

You are given a sequence A of N positive integers. Let's define "value of a

ACMath

You are given a sequence A of N positive integers. Let's define "value of a splitting" the sequence to K blocks as a sum of maximums in each of K blocks. For given K find the minimal possible value of splittings.

$N <= 10^5$

$K <= 100$

Input:	**Output:**
5 2	6
1 2 3 4 5	

→ Reply

Na2a

16 months ago, # ^ | ☆ 0

I don't think so, but I guess it can be solved by Divide And Conquer optimization.

→ Reply

vdmedragon

15 months ago, # | ☆ 0

Could you elaborate a little me more in the "Convex Hull Optimization2" and other sections for the clearer notations.

For example, You have "k" — a constant in O(kn^2). So the first dimension is of the length K and the second dimension is of the length N?

I think it would be clearer if you can write dp[n], dp[k][n] ... instead of dp[i], dp[i][j] .

Best regards,

→ Reply

anh1l1ator

5 months ago, # | ☆ 0

I don't get it why there is a $O(logN)$ depth of recursion in Divide and conquer optimization ?
Can someone explain it ?

→ Reply

Na2a

5 months ago, # ^ | ☆ ← Rev. 2 +3

Because each time range is decreased twice.

→ Reply

anh1l1ator

5 months ago, # ^ | ☆ ← Rev. 2 0

Oh, that was very trivial.

I get it now, we spend total $O(N)$ for computing the cost at each depth 2N to be specific at the last level of recursion tree.

And therefore $O(N * logN)$ is the cost of whole computation in dividing conquer scheme for relaxation.
Thanks

→ Reply

anh1l1ator

4 months ago, # | ☆ 0

Hello , I have a doubt can anyone help?

In the divide and conquer optimization ,can we always say that it is possible to use in a system where we have to minimize the sum of cost of k continuous segments(such that their union is the whole array and their intersection is null set) such that the cost of segment increases with increase in length of the segment?

I feel so we can and we can prove it using contradiction Thanks :)

→ Reply

xforceco

3 months ago, # | ☆ +5

For convex hull optimizations, with only $b[j] \geq b[j + 1]$ but WITHOUT $a[i] \leq a[i + 1]$,

I don't think the complexity can be improved to O(n), but only O(n log n) Is there any example that can show I am wrong?

→ Reply

MPeti

3 months ago, # ^ | ☆ 0

I think you're right

→ Reply

xforceco

3 months ago, # | ☆ +15

ZOJ is currently dead. For the problem "Breaking String" (Knuth opt.), please find at here

→ Reply

indy256

3 months ago, # ^ | ☆ +13

fixed

→ Reply

So_Cold

2 months ago, # | ☆ 0

please someone tell me why in convex hull optimization should be $b[j] \geq b[j + 1]$ and $a[i] \leq a[i + 1]$

in APIO'10 Commando the DP equation is

Dp[i] = -2 * a * pre_sum[j] * pre_sum[i] + pre_sum[j]^2 + Dp[j] -b * pre_sum[j] + a * pre_sum[i]^2 + b * pre_sum[i] + c

we can use convex hull trick so the line is y = A * X + B

A = -2 * a * pre_sum[j]

X = pre_sum[i]

B = pre_sum[j]^2 + Dp[j] -b * pre_sum[j]

Z = a * pre_sum[i]^2 + b * pre_sum[i] + c

and then we can add to Dp[i] += Z , because z has no relation with j

the question is , since **a** is always negative (according to the problem statement) and pre_sum[i],pre_sum[j] is always increasing we conclude that $b[j] \leq b[j + 1]$ and $a[i] \leq a[i + 1]$

I've coded it with convex hull trick and got AC , and the official solution is using convex hull trick

someone please explain to me why I'm wrong or why that is happening

thanks in advance

→ Reply

dcms2

5 weeks ago, # ^ | ☆ +8

if b[j] >= b[j + 1], then the technique is going to calculate the minimum value of the lines, if b[j] <= b[j + 1], then it's going to calculate the maximum value of the lines, as this problem requires.

→ Reply