indy256's blog

Dynamic Programming Optimizations

By indy256, 5 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	C C
Convex Hull Optimization1	$dp[i] = min_{j<i}\{dp[j] + b[j] \star a[i]\}$	$b[j] \geq b[j+1]$ ~~optionally~~ $a[i] \leq a[i+1]$	(
Convex Hull Optimization2	$dp[i][j] = min_{k<j}\{dp[i-1][k] + b[k] * a[j]\}$	$b[k] \geq b[k+1]$ ~~optionally~~ $a[j] \leq a[j+1]$	(
Divide and Conquer Optimization	$dp[i][j] = min_{k<j}\{dp[i-1][k] + C[k][j]\}$	$A[i][j] \leq A[i][j+1]$	(
Knuth Optimization	$dp[i][j] = min_{i<k<j}\{dp[i][k] + dp[k][j]\} + C[i][j]$	$A[i,j-1] \leq A[i,j] \leq A[i+1,j]$	(

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

→ Pay attention

Before contest
Codeforces Round #511 (Div. 1)
06:15:41
Register now »
*has extra registration

Before contest
Codeforces Round #511 (Div. 2)
06:15:41
Register now »
*has extra registration

Like 48 people like this. Sign Up to see what your friends like.

→ Top rated

#	User	Rating
1	tourist	3509
2	OOOOOOOOOOOOOOO... O	3327
3	Um_nik	3297
4	Syloviaely	3274
5	LHiC	3216
6	Petr	3161
7	Benq	3130
8	ko_osaga	3124
9	Swistakk	3089
10	dotorya	3060

Countries | Cities | Organizations View all →

→ Top contributors

#	User	Contrib.
1	Radewoosh	185
2	rng_58	161
3	tourist	158
4	Petr	152
5	Swistakk	150
5	Vovuh	150
7	PikMike	147
8	csacademy	146
9	Errichto	145
9	Um_nik	145

View all →

→ Find user

Handle: Find

→ Recent actions

- "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 5 years ago 71

Comments (71)

Write comment?

5 years ago, # | ← Rev. 4 **+27**

YuukaKazami

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 $a_0 < a_1 < a_2, \ldots, a_{k-1} < a_k$. 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

5 years ago, # ^ | **+3**

could you please give some example problems?
→ Reply

Secret.Codes

9 months ago, # ^ | **0**

Please give a sample problem or describe by a problem.
→ Reply

cgy4ever

5 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

5 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

21 month(s) ago, # ^ | **0**

There is both concave & convex quadrangle inequalities

PikMike → Educational Codeforces Round 51 [Rated for Div. 2]

ditoly → Codeforces Round #511

huawei → Huawei Honorcup Marathon 1

GreenLantern__1 → Veteran Competitive Programmers of Codeforces — What would you say to your younger self?

usaco_dp → DP and Graph Problems On USACO

Vovuh → Codeforces Round #510 (Div. 2) Editorial

nemeyeth → Runtime Error?

NALP → Codeforces Round #149 (Div. 2) Tutorial

lyft → The Lyft Level 5 Challenge

gKseni → ICPC World Finals 2018

hmehta → Topcoder SRM 737 — Results are Up!

achi_basadzishvili → EJOI 2018virt. contest

fcspartakm → Codeforces Round #337 (Div.2) Editorial

MikeMirzayanov → Codeforces Round #218 (Div. 2): Tutorial

cgy4ever → Codeforces Round #228 — Editorial

Arpa → [Tutorial] Everything about unordered_map

rofi93 → Help needed for SPOJ — ORDERSET

Charis02 → Educational Round 51 — Problem F Probabilistic Solution Discussion

ahmed_aly → Extra A2OJ Ladders

mostafa.saad.fci → Online Free Problem-Solving Coaching

valkyrie → Mirror of Bubble Cup 11 Finals on Codeforces

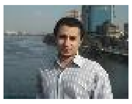
Ajosteen → Codeforces Round #509 (Div. 2, based on Qualification Stage of Southern Subregion of NEERC)

Vovuh → Codeforces Round #506 (Div. 3) Editorial

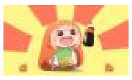
gyanendra371 → codechef question andsqr

Ajosteen → Codeforces Round #509 (Div. 2) Editorial

Detailed →

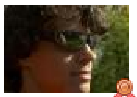

thecortex

There is both concave & convex quadrangle inequalities. concave is for minimization problems, while convex is for maximization problems. refer to Yao'82.
→ Reply


szawinis

14 months ago, # ^ | 0

How do you prove that if this condition is met, then A[i][j]<= A[i][j+1]?
→ Reply


Sammarize

5 years ago, # | +18

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

khatribiru

23 months ago, # ^ | +3

More Problem Collection.
→ Reply

kingofnumbers

5 years ago, # | +13

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

indy256

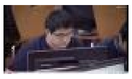
5 years ago, # ^ | +5

fixed
→ Reply

Zlobober

5 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

5 years ago, # | +8

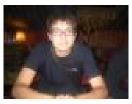
this one seemed a nice dp with optimization to me:https://www.hackerrank.com/contests/monthly/challenges/alien-languages
→ Reply

hogloid

5 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

5 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[]

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

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

→ Reply

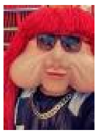
victorsenam

3 years ago, # ^ | ← Rev. 2 0

GOODG can be solved with Type 1

EDIT: I explain that below.

→ Reply


fofao_funk

2 years 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 how to apply the "type 1" optimization.

→ Reply

samier_aldroubi

2 years 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

14 months ago, # ^ | ← Rev. 2 +13

Yeah, I'm sorry about saying this and not explaining. Actually I should give credit because ItsYanBitches first realized the fully dynamic approach was not necessary. Here's my code.

Maybe the most natural approach for this problem is to try to solve the following recurrence (or something similar) where $f(0) = 0$ and $d_0 = 0$:

$$f(i) = max_{j < i}(f(j) - d_j * (i - j)) + a_i$$

Well, this recurrence really requires a fully dynamic approach. We'll find one that doesn't. Instead of trying to solve the problem for each prefix, let's try to solve it for each suffix. We'll set $g(n + 1) = 0$, $a_0 = d_0 = 0$ and compute

$$g(i) = max_{j > i}(g(j) - d_i * (j - i) + a_j)$$

which can be written as

$$g(i) = max_{j > i}(- d_i * j + a_j + g(j)) + d_i * i)$$

now we notice that the function inside the max is actually a line with angular coefficient j and constant term $a_j + g(j)$ (which are constant on i) evaluated at $- d_i$. Apply convex trick there (the standart one) and we're done.

Notifying possibly interested people after a long delay (sorry about that again): **fofao_funk**, **samier_aldroubi** and **synxazox**. And sorry in advance for any mistake, the ideia for the solution is there.

→ Reply

victorsenam

14 months ago, # ^ | +3

Why the downvotes? Is it wrong?

→ Reply

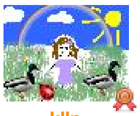
3 years ago, # ^ | +3

New link for Commando:

victorsenam

New link for Commando:

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


kllp

5 years ago, # | 0

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

indy256

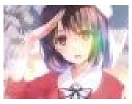
5 years ago, # ^ | +4

Try to zoom out, pressing Ctrl + -
→ Reply

Monyura

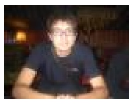
5 years ago, # | ← Rev. 2 +8

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

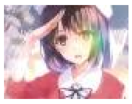
4 years ago, # | 0

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

4 years ago, # ^ | +1

Check this one : Guardians of the Lunatics
→ Reply


mbrc

4 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

3 years 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

micklepru

3 years 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.

Check out my submission.
→ Reply

92anurag

4 years ago, # | +3

can anyone provide me good editorial for dp with bitmask .
→ Reply

4 years ago, # | 0

Has matrix-exponent optimizations been included here?
→ Reply

D_puongs

matthew69

7 months ago, # ^ | 0

wtf do you mean by matrix-expo optimization?

→ Reply

Farsid

4 years ago, # | +2

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

→ Reply

indy256

4 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.

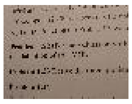
→ Reply

Thomas_Ahle

3 years ago, # ^ | 0

Link to the Hu and Shing algorithm?

→ Reply


newCEA

2 years 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.

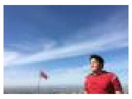
→ Reply

mayankp

4 years ago, # | +1

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

→ Reply


Conqueror

3 years 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 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.

$N <= 10^5$

$K <= 100$

K <= 100

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

→ Reply

Na2a

3 years ago, # ^ | ▲ 0 ▼

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

TimonKnigge

9 months ago, # ^ | ← Rev. 3 ▲ 0 ▼

Divide and Conquer optimization doesn't work here since the monotonicity of the argmin doesn't hold, consider e.g. `2 3 1 5`. The optimal partition is `[2] [3 1 5]` but when you remove the `5` it becomes `[2 3] [1]`.
→ Reply

vdmedragon

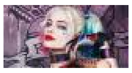
3 years ago, # | ▲ 0 ▼

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

2 years 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

2 years ago, # ^ | ← Rev. 2 ▲ +3 ▼

Because each time range is decreased twice.
→ Reply

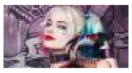
anh1l1ator

2 years 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

2 years 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 :)

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

xforceco

2 years ago, # | +5

For convex hull optimizations, with only b[j]≥b[j+1] but WITHOUT a[i]≤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

2 years ago, # ^ | 0

I think you're right
→ Reply

xforceco

2 years ago, # | +15

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

indy256

2 years ago, # ^ | +13

fixed
→ Reply

So_Cold

2 years ago, # | 0

please someone tell me why in convex hull optimization should be b[j]≥b[j+1] and a[i]≤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] ≤ b[j+1] and a[i]≤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

2 years 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

satyaki3794

21 month(s) ago, # | +10

Is it necessary for the recurrence relation to be of the specific form in the table for Knuth's optimization to be applicable? For example, take this problem. The editorial mentions Knuth Optimization as a solution but the recurrence is not of the form in the table. Rather, it is similar to the Divide-and-Conquer one, i.e. dp[i][j]=mink<j{dp[i-1][k]+C[k][j]}. Does anyone know how/why Knuth's optimization is applicable here?
→ Reply

13 months ago, # | ← Rev. 2 +3

It is also worthwhile to mention the DP Optimization given here sdnr1

It is also worthwhile to mention the DP Optimization given here http://codeforces.com/blog/entry/49691 in this post.
→ Reply

Straw

11 months ago, # |
0
Can we have the same dp optimizations with dp[i][j] = max (dp....)?
→ Reply

P__

11 months ago, # ^ |
0
Yes.
→ Reply

Straw

10 months ago, # ^ |
0
In that case (dp[i][j] = max (dp...)) the condition still unchanged : A[i][j]≤A[i][j+1]. Is that true? Thanks!
→ Reply

P__

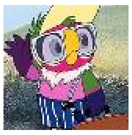
10 months ago, # ^ |
0
Yes.
→ Reply

neeraj745

8 months ago, # |
0
can someone please give me intuition on proof of A[i,j-1]≤A[i,j]≤A[i+1,j] given for knuth optimization of optimal binary search tree.
→ Reply

anupamwadhwa

7 months ago, # |
0
You can try **LARMY** on spoj for Divide and Conquer Optimization.
→ Reply


sslotin

6 months ago, # |
+5

$dp[i,j] = \min_{k<j}\{dp[i-1,k] + c[k][j]\}$
Okay gang let's see who this recurrence really is.
$dp[i,j] = \min_{k<j}\{dp[i-1,k] + c[k][j]\}$
$dp[i,j] = \min_{\vec{a}\in S}\{\vec{a}\cdot\vec{b}\}$
Convex hull maintenance ?!!

→ Reply

bktl1love

5 months ago, # | 0

Fresh problem. Can be solved with CHT.

→ Reply

Expelliarmus123

4 months ago, # | +8

What will be the actual complexity of Knuth optimization on a O(n^2 * k) DP solution ? Will it be O(n^2) or O(n*k)? Here n = number of elements and k = number of partitions.

→ Reply

Expelliarmus123

4 months ago, # ^ | +13

I would like to mention **Zlobober**, **rng_58**, **Errichto**, **Swistakk** here politely. It would be really helpful if you kindly answered my query.

→ Reply

Errichto

4 months ago, # ^ | +3

It can only remove n from the complexity because it (more or less) gets rid of iterating over all elements of the interval (up to n elements).

→ Reply

4 months ago, # ^ | +8

Expelliarmus123

Thanks a lot for replying to me. So what you are saying is a O(n^2 * k) solution will turn into a O(n * k) solution? I got TLE in problem SPOJ NKLEAVES (n=10^5, k=10) using Knuth optimization, but passed comfortably using divide & conquer optimization in O(n k log n). That's why I am curious to know whether knuth optimization reduces a n factor or a k factor from the original O(n^2 * k) solution, since a O(n*k) solution should have definitely passed. Would you please have a look?

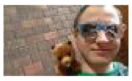
→ Reply

Errichto

4 months ago, # ^ | +3

Can you show the code? I'm not sure but I think that Knuth optimization can be used here to speed up the $O(n^3 \cdot k)$ solution with states $dp[left][right][k]$. The improved complexity would be $O(n^2 \cdot k)$.

→ Reply



Swistakk

4 months ago, # ^ | -10

How does it help in any way xd?

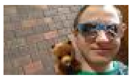
→ Reply

Errichto

4 months ago, # ^ | +3

He asked if N or K will be removed from the complexity, I answered. What is wrong here?

→ Reply



Swistakk

4 months ago, # ^ | -10

You said "at most n" what brings 0 bits of information since k<=n.

→ Reply

Zlobober

4 months ago, # ^ | +13

It is $O(n^2)$ as the complexity upper bound is proven by summating the running time of DP value calculation over the diagonals of the $n \times k$ DP matrix. There are $n + k - 1$ diagonals, on each of them running time is at most $O(n)$ due to the given inequalities, so the total running time is $O((n+k)n) = O(n^2)$.

→ Reply

DGC

4 months ago, # | ← Rev. 2 0

Someone know where i can find an article about **Lagrange optimization**? (i know that this can be used for reduce one state of the `dp` performing a binary search on a constant and add this on every transition until the realized number of transition for reach the final state become the desired)

→ Reply