indy256's blog

Dynamic Programming Optimizations

By indy256, 17 months 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	Original Complexity	Optimized Complexity	Links
Convex Hull Optimization1	$dp[i] = min_{j<i}\{dp[j] + b[j] \star a[i]\}$	$b[j] \geq b[j+1]$ optionally $a[i] \leq a[i+1]$	$O(n^2)$	$O(n)$	1 p1
Convex Hull Optimization2	$dp[i][j] = min_{k<j}\{dp[i-1][k] + b[k] * a[j]\}$	$b[k] \geq b[k+1]$ optionally $a[j] \leq a[j+1]$	$O(kn^2)$	$O(kn)$	1 p1 p2
Divide and Conquer Optimization	$dp[i][j] = min_{k<j}\{dp[i-1][k] + C[k][j]\}$	$A[i][j] \leq A[i][j+1]$	$O(kn^2)$	$O(knlogn)$	1 p1
Knuth Optimization	$dp[i][j] = min_{i<k<j}\{dp[i][k] + dp[k][j]\} + C[i][j]$	$A[i,j-1] \leq A[i,j] \leq A[i+1,j]$	$O(n^3)$	$O(n^2)$	1 2

					p1

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**: $C[a][c] + C[b][d] \le C[a][d] + C[b][c],\ a \le b \le c \le d$
- **monotonicity**: $C[b][c] \le C[a][d],\ a \le b \le c \le d$
- 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 17 months ago 21

→ Top contributors

#	User	Contrib.
1	Fefer_Ivan	145
2	DmitriyH	140
2	Petr	140
4	marat.snowbear	131
5	cgy4ever	130
6	Xellos	129
7	LeBron	128
8	skyvn97	127
9	SkidanovAlex	124
9	RodionGork	124
9	BYN	124

View all →

→ Find user

Handle:

Find

→ Recent actions

caioaao → ACM ICPC 2014 Latin America Finals

laoriu → Codeforces Round #277(Div. 2) Editorial

MikeMirzayanov → Codeforces Round #277.5 (Div. 2)

joker99 → Need help on Recursion !!!

Comments (21)

Write comment?

17 months ago, # | ← Rev. 4 +27

WJMZBMR

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

17 months ago, # ^ | +3

could you please give some example problems?

→ Reply

adi14 → Can we perform a large number of Range Update and Point Update Queries in less than Amortized log(n) time?

eddycael → How to calculate modular multiplicative inverse with no prime numbers?

saba_tavdgiridze → MLE

Bugman → Codeforces Round #276 — Editorial

MLGuy → [ACM] Bangkok regional 2014 result

LaughIsLife → SPOJ-QTREE2

Fefer_Ivan → All Andrew Stankevich Contests at Gym

Emruz_Hossain → How I can overcome my fear about math?

trungpham90 → Java version

Alim14 → Small Code Library for Beginner

MikeMirzayanov → 2014-2015 CT S02E09: Codeforces Trainings Season 2 Episode 9

hexor → T.O.I(Ubilo)

RodionGork → Programming as Art or Craft

marat.snowbear → A job for the red one

csolvekorb → LCS

MikeMirzayanov → 10 Reasons to Prepare Problems in Polygon

MikeMirzayanov → 2014-2015 CT S02E04: Codeforces Trainings Season 2 Episode 4

algorithmov → Range Minimum Query using sparse tables

justHusam → How to use array instead of