CODEFORCES
Sponsored by Telegram

|
JacobianDet | Logout

HOME CONTESTS GYM PROBLEMSET GROUPS RATING API CALENDAR

TOURIST BLOG TEAMS SUBMISSIONS GROUPS CONTESTS PROBLEMSETTING

tourist's blog

Hello 2018 -- Tutorial

By **tourist**, history, 3 days ago, translation,

Here is the tutorial of Hello 2018. Enjoy!

Modular Exponentiation
Christmas Spruce
Party Lemonade

Problem writer: **tourist**

913C - Party Lemonade

Note that if $2 \cdot a_i \le a_{i+1}$, then it doesn't make sense to buy any bottles of type $i+1$ — it won't ever be worse to buy two bottles of type i instead. In this case let's assume that we actually have an option to buy a bottle of type $i+1$ at the cost of $2 \cdot a_i$ and replace a_{i+1} with $min(a_{i+1}, 2 \cdot a_i)$. Let's do this for all i from 1 to $n-1$ in increasing order.

Now for all i it's true that $2 \cdot a_i \ge a_{i+1}$. Note that now it doesn't make sense to buy more than one bottle of type i if $i < n$. Indeed, in this case it won't ever be worse to buy a bottle of type $i+1$ instead of two bottles of type i. From now on, we'll only search for options where we buy at most one bottle of every type except the last one.

Suppose that we had to buy exactly L liters of lemonade, as opposed to at least L. Note that in this case the last $n-1$ bits of L uniquely determine which bottles of types less than n we have to buy. Indeed, if L is odd, then we have to buy a bottle of type 0, otherwise we can't do that. By the same line of thought, it's easy to see that bit j in the binary representation of L is responsible for whether we should buy a bottle of type j. Finally, we have to buy exactly $\lfloor L / 2^{n-1} \rfloor$ bottles of type n.

But what to do with the fact that we're allowed to buy more than L liters? Suppose we buy $M > L$ liters. Consider the highest bit j in which M and L differ. Since $M > L$, the j-th bit in M is 1, and the j-th bit in L is 0. But then all bits lower than the j-th in M are 0 in the optimal answer, since these bits are responsible for the "extra" bottles — those for which we spend money for some reason, but without which we would still have $M > L$.

Thus, here is the overall solution. Loop over the highest bit j in which M differs from L. Form the value of M, taking bits higher than the j-th from L, setting the j-th bit in M to 1, and bits lower than the j-th to 0. Calculate the amount of money we have to pay to buy exactly M liters of lemonade. Take the minimum over all j.

The complexity of the solution is $O(n)$ or $O(n^2)$, depending on the implementation.

Solution

```
#include <bits/stdc++.h>

using namespace std;

int main() {
  int n, L;
  scanf("%d %d", &n, &L);
  vector<int> c(n);
  for (int i = 0; i < n; i++) {
    scanf("%d", &c[i]);
  }
```

→ Pay attention

Before contest
Educational Codeforces Round 36 (Rated for Div. 2)
39:42:19

Like 175 people like this. Sign Up to see what your friends like.

→ JacobianDet

Rating: 1244
Contribution: 0

- Settings
- Blog
- Teams
- Submissions
- Talks
- Contests

JacobianDet

→ Top rated

#	User	Rating
1	tourist	3496
2	Um_nik	3309
3	Petr	3298
4	dotorya	3225
5	W4yneb0t	3218
6	OO0OOO00O0OOO0O0...O	3199
7	Radewoosh	3150
8	izrak	3109
9	anta	3106
10	ershov.stanislav	3105

Countries | Cities | Organizations View all →

→ Top contributors

#	User	Contrib.
1	tourist	182
2	rng_58	176
3	csacademy	167
4	Petr	161
5	Swistakk	156
6	Errichto	150
7	lewin	149
8	Zlobober	145
9	matthew99	141
9	adamant	141

View all →

→ Find user

Handle:
Find

```
  for (int i = 0; i < n - 1; i++) {
    c[i + 1] = min(c[i + 1], 2 * c[i]);
  }
  long long ans = (long long) 4e18;
  long long sum = 0;
  for (int i = n - 1; i >= 0; i--) {
    int need = L / (1 << i);
    sum += (long long) need * c[i];
    L -= need << i;
    ans = min(ans, sum + (L > 0) * c[i]);
  }
  cout << ans << endl;
  return 0;
}
```

Too Easy Problems
Logical Expression
Strongly Connected Tournament
Power Substring
Don't Exceed

Tutorial of Hello 2018

+414 tourist 3 days ago 99

Comments (99)

Write comment?


math314

3 days ago, # | +7

Not a single tutorial is available ? Only solutions ?

→ Reply


tourist

3 days ago, # ^ | +144

Sorry, the tutorials will be available shortly after the end of system testing.

→ Reply

I_LOVE_METSUKA

3 days ago, # ^ | +43

Don't be sorry man. Forgive us for being stupid and not understanding you problems, sorry.

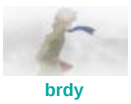
→ Reply


inaderi268

35 hours ago, # ^ | +8

sag salam miresoone

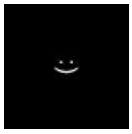
→ Reply


brdy

3 days ago, # | +1

I'm liking this new trend.

→ Reply


Paradosk

3 days ago, # | 0

why this solution worked? cin>>n>>m; int a=pow(2,n); cout<<m%a<<endl; what about overflow?!

→ Reply


Coroian_David

3 days ago, # ^ | 0

a will be -2147483648, but in the way C++ works with negative modulo it will work...

→ Reply

→ Recent actions

RockyB → IZHO 2018

gepardo → Sqrt-tree: answering queries in O(1) with O(NloglogN) preprocessing.

DarthPrince → Algorithm Gym :: Graph Algorithms

csacademy → Special Round #64 (Interactive only) with prizes

shawonchy → 287_B_Pipelines(binary search)

Med0b1011 → Topcoder SRM 727 By Errichto

tourist → Hello 2018 -- Tutorial

BledDest → Educational Codeforces Round 35 — Editorial

-synx- → Mersenne Prime — M61 for Hashing

lewin → Good Bye 2017 Editorial

Sla7er → Invitation for CODE-IT 2018

tourist → Hello 2018

xuanquang1999 → C++14 weird behavior

Z38 → C++11 Build system for Sublime text

chari407 → Facebook Hacker Cup 2018

PraveenDhinwa → Invitation to ACM-ICPC Asia-Kharagpur Onsite Replay Contest On CodeChef

Snezhok → Codeforces Round #450(Div. 2). Editorial.

Ehsan_sShuvo → Need problems on BitMask

tusu → IZHO 2018

Al.Cash → Efficient and easy segment trees

ivplay → getting WA need help (lighoj1390 - Weight Comparison) (Category: Cycles, Topological Sorting, Strongly Connected Component)

Hamzqq9 → Strange verdict in compare function while sorting

snarknews → New Year Contests from SnarkNews

Hikari9 → Easy geometry using std::complex

GiantSlayer → Help Post

Detailed →

Reply

yassin_

3 days ago, # ^ | ☆ ← Rev. 2 0

a could be any value, couldn't it? Depending on n. It's unlikely to be <= 10^8, but possible or not?

→ Reply

Coroian_David

3 days ago, # ^ | ☆ ← Rev. 3 0

It varies depending on the PC architecture (32 or 64 bits), compiler and operating system, in my PC and in Codeforces custom invocation it has that value. For every n >= 31 it has the same value.

→ Reply

Arturoku

3 days ago, # ^ | ☆ 0

Even though it overflows the % operation is unsigned so the number remains big enough to work properly.

→ Reply

abhishek125

3 days ago, # ^ | ☆ 0

actually i made one unsuccessful hack attempt because i didn't know how c++ work with modulo operator.you can read how c++ works with modulo operator here.

→ Reply

Dodooooo

3 days ago, # ^ | ☆ ← Rev. 2 -10

in c++ when a large double number (larger than the receiver type) is put into smaller receiver it automatically casts it to the largest possible number to be put in the receiver in this case INT_MAX so when a number is calculated % INT_MAX it will always give the number

→ Reply

yassin_

3 days ago, # ^ | ☆ +5

Can you give a source for this information?

→ Reply

Dodooooo

2 days ago, # ^ | ☆ -7

you can try it write down cout << (int)1e12 and write down cout << INT_MAX see the results of the two operations .

→ Reply

yassin_

2 days ago, # ^ | ☆ +5

That is no proof, could be undefined behavior

→ Reply

Dodooooo

2 days ago, # ^ | ☆ Rev. 2 0

it could be undefined behavior but if it's the case then it should change from one platform to another it produces the result as I said in Clion, codeforces platform and these other online IDEs

https://www.ideone.com/A37CME

https://onlinegdb.com/Sy1f-DfNz

http://cpp.sh/5frra

http://tpcg.io/bnedib

you can check it

you can check it

→ Reply

yassin_

2 days ago, # ^ | ☆ ← Rev. 2 0

First of all, it depends on the compiler, not the IDE. Then also it *might* change on a different platform, but it doesn't have to.

edit: I also found this, which says it's undefined behavior: https://stackoverflow.com/questions/526070/handling-overflow-when-casting-doubles-to-integers-in-c

→ Reply

Dodooooo

2 days ago, # ^ | ☆ Rev. 2 0

if it's the same on different platforms , then how could it be undefined ? also is all these IDEs use the same compiler ?

yes it's indeed undefined in the older versions you can try it in c++98 or c++11 it won't be like that

→ Reply

kwangg

47 hours ago, # ^ | ☆ +5

undefined means that it's not specified in the C++ standard what to do for that scenario. Maybe all compilers implement the same solution, maybe some do it differently from others.

→ Reply

Dodooooo

47 hours ago, # ^ | ☆ 0

thanks a lot kwangg

→ Reply

mouse_wireless

2 days ago, # ^ | ☆ +5

The standard specifications mention it is indeed undefined behavior (read here). In practice, modern compilers do indeed cast large float numbers to the maximal value acceptable, when cast to an integral type.

→ Reply

Franklyn_W

3 days ago, # | ☆ +3

I think on F, that ans[0] = ans[1] = 0..

Thanks for the contest, it was fun!

→ Reply

YakutovDmitriy

3 days ago, # ^ | ☆ +4

Thank you, tutorial was updated, changes will come into force soon.

→ Reply

architj

3 days ago, # | ☆ ← Rev. 2 -16

I used a **greedy** solution for question C .First I found out the bottle which has least value of cost per litre .Then I used a formula to give maximum possible number of bottles of this category(i.e. L/(volume of bottle of the category). Then i **brute forced** my solution (i bought the best possible bottle at the moment by using a formula and then subtracted the volume from L, then repeat and repeat till L becomes <= 0). However, my **solution failed** on test case 14 of system testing. Here is my solution link My solution Any help would be appreciated. Thanks :)

→ Reply

shubhiks1032

3 days ago, # ^ | ☆ +12

It fails for the case:

Spoiler 1

Spoiler 2

Spoiler 3

→ Reply

Shudi

3 days ago, # ^ | ☆ 0

I used quite similar **greedy** approach for problem C with one little modification and I got AC. Maybe someone will find it interesting. Firstly, let's sort the bottles by the cost per litre. Let's start from the cheapest of them going to the most expensive. Lets denote the size of the bottle *i* as V[i]. We have two different cases:

- **V[i] <= L**. Here we have the best price because we going down from the cheapest bottle and we will buy all the V[i] litres. Then we can add to the answer this price and multiply that by the number of times. Of course we should subtract this V[i] multiplied by the number of times from the *L*. Then we can buy one more bottle with this size and it will be more litres than we need, but maybe it has the cheapest price and it will be the best variant? Otherwise, we can continue going down to the more expensive bottles per litre, but we will not buy the extra litres as it was before. We need just to update our realAns as you can see it in the code below:

```
if(a[i].V <= L) {
    ll times = (L / a[i].V);
    ans += times * a[i].cost;

    L -= a[i].V * times;
```

```
    realAns = min(realAns, ans + a[i].cost);
}
```

- **V[i] > L**. We can buy this bottle and it will be our ans. Here we go:

```
realAns = min(realAns, ans + a[i].cost);
```

This solution seems nice to me. Here the it is 34026125 . Feel free to ask questions if something is not clear for you.
→ Reply

3 days ago, # ^ | ☆ ← Rev. 2 0

3 7 5 11 18

HOMIARA_RUBY

are you sure your your code gives correct ans for this case? If yes then how?

UPD: GOT IT. I WAS WRONG.
→ Reply

3 days ago, # ^ | ☆ ← Rev. 2 0

mtarunsai1999

it fails when the volume of first bottle(one with cheapest price per liter) is greater than l and buying that bottle is the cheapest option. You logic skips that possibility. In that case, the solution is min(price of this bottle, cost evaluated without considering this bottle).
→ Reply

3 days ago, # | ☆ -15

Le_Savage

Great set of problems.I liked the 3rd one most.Credits goes to @tourist, my man. Thumbs up for all the problem setters too.
→ Reply

3 days ago, # | ☆ 0

abh_u

Is there different kind of solution for problem C?
→ Reply

3 days ago, # ^ | ☆ ← Rev. 2 0

WDragon

Here, 34034532 .

Same logic but a bit different implementation. :)
→ Reply

3 days ago, # ^ | ☆ 0

kwangg

The solution I wrote only passes through the bits of L once, going from least-significant to most-significant. 34027236

If, at any time, buying bottle i would be better than the entirety of the sum we've already calculated, then replace the sum with the cost for bottle i.
→ Reply

3 days ago, # | ☆ +8

hm_98

E problem uses the same concept as F Problem of this contest.
→ Reply

3 days ago, # ^ | ☆ +3

chemthan

I think E is suitable for smart or patient person. I quit immediately after reading it.
→ Reply

3 days ago, # | ☆ +18

Misread problem D. How to solve the problem optimally when k >= a_i instead of

mishraiiit

Misread problem D. How to solve the problem optimally when k >= a_{p_j} instead of k ≤ a_{p_j} in problem D?

→ Reply

3 days ago, # ^ | ☆ ← Rev. 2 +6

I think that the beauty of problem D comes when you think of it geometrically. Imagine that you have a point in the plane with coordinates $(a_i, t_i) \forall 1 \leq i \leq n$.

Guty

Then, for a fixed k, they are asking you a question about the points in a semiplane that lies to the right of the vertical line $x = k$, including the line (this is in the original statement with $k \geq a_{p_j}$, it lies to the left of that line if we choose $k \leq a_{p_j}$).

With this in mind, I think that both problems look more like a "Sweep Line" kind of task, doesn't it? (this is how I approached problem D during the contest, and why I believe that solution 2 in the editorial is more natural, although this last part is clearly subjective)

→ Reply

mishraiiit

3 days ago, # ^ | ☆ ← Rev. 4 +3

"it lies to the left of that line if we choose ....", not exactly. When we count problems in score such that k >= a_{p_j}, we can choose problems on either side of the line. The points for which k >= a_{p_j} are added to the score, however, we can choose points on other side of the line which don't satisfy this condition which will increase value of k (not increasing the score though), however this also moves the line and we can add new points with lesser ti such that k >= a_{p_z} increases and we can replace few points.

Basically, when k >= a_{p_k}, the final points/score will not always be equal to the number of problems we choose.

→ Reply

AleksanderBalobanov

32 hours ago, # ^ | ☆ ← Rev. 2 +5

I misread it too and used the following idea. Let's sort our arrays in non-descending order of a_i. Run a binary search on the value of x result points, and then say that the total amount of solved problems should be at least a_i. Then for any $j <= i$ j-th element can give us a point if we will take it and we will suppose now that all the elements right to the i-th will not give us a point(if they actually will we'll consider it later). So we should take x elements from the i - th prefix with minimum total time, and then we should take a_i - x elements from all the array with minimum total time(except the taken ones). It also can be noticed(but it is not neccesary) that this check procedure for a given x is to find out whether there exists such i that the sum of minimum x elements from the i-th prefix plus sum of minimum a_i - x elements from the $i + 1$-th suffix is no more than T. It can be implemented with two segment trees(for prefix and suffix elements) (where trees are build on the array sorted by the time now and making a transition from i to $i + 1$ we do some changes, and find the sum of some first values).

→ Reply

chemthan

3 days ago, # | ☆ +57

My rating is now fibonacci sequence (2358) :)

→ Reply

3 days ago, # | ☆ 0

Can someone find out why this code for D Too Easy Problems (based on binary

kverti98

search) get runtime error?
http://codeforces.com/contest/913/submission/34026112 Thanks in advance.
→ Reply

JustLive

3 days ago, # ^ | ☆ 0

http://codeforces.com/contest/913/submission/34035369

In your cmp function change <= to <
→ Reply

kverti98

3 days ago, # ^ | ☆ 0

How can it cause runtime error? Is it about implementation of std::sort? Thank you for finding error :D
→ Reply

JustLive

3 days ago, # ^ | ☆ +9

The reason is cmp(a, b) should return true only when a is strictly less than b (a < b). It should return false when elements are equal (a == b). If you use <= then for case a == b it will return true for both cmp(a, b) and cmp(b, a). For complier it means a < b and b < a at the same time so you will probably end up in indefinite loop.

Some links:
http://en.cppreference.com/w/cpp/concept/Compare and http://codeforces.com/blog/entry/54610 (item 3)
→ Reply

soohotiam

3 days ago, # | ☆ +5

Thank You [user:tourist]for nice and pretty problem. Now I am a Blue Coder :'(. I am Very Happy For That. :)
→ Reply

shubhiks1032

3 days ago, # | ☆ +20

Its my first time seeing an $\mathit{Integral}$ being applied in competitive programming (in problem H)! :3
→ Reply

mouse_wireless

3 days ago, # ^ | ☆ ← Rev. 2 0

There is this very interesting problem that uses integrals (now that I read H from this contest, the problems are *somewhat* similar).

EDIT: ok, maybe not that similar since in the problem I pointed out you could calculate the integral with general algorithms such as Riemann integration or Simpson's rule, whereas here you have to mathematically compute the integral since you need an exact answer.
→ Reply

fchirica

3 days ago, # ^ | ☆ +5

http://main.edu.pl/en/archive/oi/11/zga

This one has integrals and it's also about gambling (a nice bonus of the problem!).
→ Reply

apar03

3 days ago, # | ☆ +5

Solution to C using Binary Search:
http://codeforces.com/contest/913/submission/34035650
→ Reply

3 days ago, # ^ | ☆ 0

Can you explain your approach? Thanks!

Ewwa

Can you explain your approach? Thanks!

→ Reply

apar03

3 days ago, # ^ | ☆ +14

Maintain a vector for the lemonade types and sort it according to the cost per liter(c[i]/(2^(i-1)). This sorts the lemonades according to the most optimal types of lemonade (cheapest lemonade) by normalizing the quantity to 1 liter. Thus we can greedily say that, if we buy lemonades in this sorted order, we can obtain maximum liters of lemonade from a given amount of money.

Therefore, now we can use binary search to fix the amount of money we have, and then see if it is possible to obtain atleast L liters of lemonade by traversing in this sorted order and taking as many no. of lemonades of each type as possible.

Hence, amount of money used can be easily minimized.

→ Reply

HOMIARA_RUBY

3 days ago, # ^ | ☆ ← Rev. 2 0

3 7 5 11 18

what about this case? are you sure your code passes this case? UPD: Got it. I was wrong.

→ Reply

HOMIARA_RUBY

3 days ago, # ^ | ☆ 0

ok. I checked this case against LGM's code. Your code gives similar to them. But how its 33? shouldn't it be 34?

→ Reply

apar03

3 days ago, # ^ | ☆ 0

Buy 3 one liter bottles, cost=3*5=15 Buy 1 four liter bottle, cost=18

Net cost=15+18=33 to buy 7 liters of lemonade

→ Reply

vivace_jr

3 days ago, # | ☆ +5

In problem C purpose of doing the following is? L -= need << i;

→ Reply

rocky8086

20 hours ago, # ^ | ☆ +1

you are taking "need" number of bottles of 2^i size. "need << i" is basically "need * (2^i)"

→ Reply

vivace_jr

13 hours ago, # ^ | ☆ 0

got it,thanks

→ Reply

CODENT47

3 days ago, # | ☆ ← Rev. 2 0

Can anyone tell why my code for **C** is failing for last sample input. The logic is similar to editorial. Link to my solution : My Solution

→ Reply

3 days ago, # ^ | ☆ +4

See for example the following test case:

Jonno

See for example the following test case:

5 7

1 2 3 4 9

The correct output is 4 (take one bottle of 8 liters) but your answer gives 6.

Hope this helps you!

→ Reply

CODENT47

3 days ago, # ^ | ☆ +3

Thanks a lot! Understood my mistake

→ Reply

ZaakDov

3 days ago, # | ☆ 0

for problem G,the tutorial says "In this problem n ≤ 11. m = 6 is enough for such n",why 6 is enough?

→ Reply

YakutovDmitriy

3 days ago, # ^ | ☆ 0

$m = 6$ is enough because if $n \le 11$ then $10^m \ge 2 \cdot 2^{n+m}$.

Let $x = a \cdot 10^m$. If x is not divisible by 2^{n+m}, we add $2^{n+m} - x \bmod 2^{n+m}$ to x. After that if x is divisible by 5 we add 2^{n+m} to x.

After such transformations x is divisible by 2^{n+m}, x is not divisible by 5 and $x \le a \cdot 10^m + 2^{n+m} - 1 + 2^{n+m}$. We know that $10^m \ge 2 \cdot 2^{n+m}$. Therefore $b = x - a \cdot 10^m < 10^m$. That's exactly what we want.

→ Reply

ZaakDov

3 days ago, # ^ | ☆ ← Rev. 4 0

thank you! I didn't understand clearly why "10^m ≥ 2·2^(n + m)" <=> "there exists k satisfying equation (1) and (2)" earlier, and after read your tutorial more carefully, I got that "Consider equation . Every number in the equation is divisible by 2^(n + m), lets divide all the numbers by this. The result is — equation (3) where . Use the following lemma: Lemma For every positive integer t number 2 is primitive root modulo 5^t." shows why "10^m ≥ 2·2^(n + m)"<=>"there exists k satisfying equation (1) and (2)", and we just need to find any k under these assumes. In this way the range of an available m can be determined, very nice problem!

→ Reply

ZaakDov

3 days ago, # ^ | ☆ 0

when 10^m>=2^(n+m+1), then b can choose from [0,10^m-1], since 2^(n+m)-1<2^(n+m+1)-1<=10^m-1, so the values of b can cover [0,2^(n+m)-1], thus there exists some b satisfying ai*10^m+b=0(mod 2^(n+m)) and let x=ai*10^m+b, so x/2^(n+m)=2^(k-n-m)(mod 5^(n+m)), since 2 is a primitive root of 5^(n+m), so there exists t in range[0,4*5^(n+m-1)) satisfying 2^t=x/2^(n+m)(mod 5^(n+m)), then let k=n+m+t>=n+m, this k is the answer to the problem

→ Reply

ZaakDov

3 days ago, # ^ | ☆ 0

a way to prove 2 is primitive root of 5^k (k>=1) using induction: assuming (I) 2 is primitive root of 5^k, and (II) 2^phi(5^k)≠1(mod 5^(k+1)) first, about (II) 2^phi(5^(k+1))-1=2^5phi(5^k)-1= (2^phi(5^k)-1)* (2^4phi(5^k)+2^3phi(5^k)+2^2phi(5^k)+2^phi(5^k)+1) since 2^phi(5^k)=1(mod5^k) (using euler's theorem) and 2^phi(5^k)≠1(mod5^(k+1)) (by assuming),therefore gcd(5^(k+2),2^phi(5^k)-1)=5^k if 5^(k+2)|2^phi(5^(k+1))-1 then there

must be 5^2|2^4phi(5^k)+2^3phi(5^k)+2^2phi(5^k)+2^phi(5^k)+1 but

2^phi(5^2)=1(mod 25), thus 2^(sphi(5^k))=(2^phi(5^2))^(s*5^(k-2))=1(mod 25) so 2^4phi(5^k)+2^3phi(5^k)+2^2phi(5^k)+2^phi(5^k)+1=5(mod25) so 2^phi(5^(k+1))≠1(mod 5^(k+2))

then about (I) for t in range [0,phi(5^(k+1))), 2^t=1(mod5^(k+1))=>2^t=1(mod5^k)=>phi(5^k)|t let t=phi(5^k)*s, since 0<=t=phi(5^k)*s<phi(5^(k+1)) so 0<=s<5, consider 2^t-1=(2^phi(5^k)-1)(2^(s-1)phi(5^k)+2^(s-2)phi(5^k)+....+1) so 5^(k+1)|2^t-1 <=> 5|2^(s-1)phi(5^k)+2^(s-2)phi(5^k)+....+1 since 2^phi(5^k)=(2^4)^(5^(k-1))=1(mod5) so 2^sphi(5^k)=(2^phi(5^k))^s=1(mod5) so 2^(s-1)phi(5^k)+2^(s-2)phi(5^k)+....+1=s(mod5) thus 5|s and 0<=s<5 => s=0 thus for t in range [0,phi(5^(k+1))),2^t=1(mod5^(k+1))=>t=0, implying that 2 is primitive root of 5^(k+1)

→ Reply

ZaakDov

3 days ago, # ^ | ☆ 0

somewhat tedious

→ Reply

nhouyng

3 days ago, # | ☆ 0

Great ur solution :)

→ Reply

xish

3 days ago, # | ☆ 0

I made two mistakes in Problem C during the contest. 34026884 First doesn't deal with a empty tail at last. Second fail to score the status and made a lot of unnecessary calculation which made TLE. My solution is too complex in all. Any suggestion in shorten the code?34038638

→ Reply

folcotandiono

3 days ago, # | ☆ +5

For problem C fourth example, why i get 44981600797903355?

→ Reply

adityakumar3811

3 days ago, # ^ | ☆ +5

I was getting that too. You must have some index of array which is getting negative.

→ Reply

folcotandiono

3 days ago, # ^ | ☆ +5

No, my solution is wrong.. Thanks for commented

→ Reply

dynamite_iit

3 days ago, # | ☆ 0

Somebody please explain for Problem E,How the number of functions are 256 and number of states are 3256?

→ Reply

PrimeG

3 days ago, # | ☆ +1

i use dfs to solve the problem C 34018956

→ Reply

badSHAH

3 days ago, # | ☆ +6

for problem 913D- Too Easy Problems

"The first observation is that if we solve a problem which doesn't bring us any points, we could as well ignore it and that won't make our result worse. Therefore, there exists an answer in which all problems bring us points. Let's consider only such answers from now on."

I don't think this is true

I don't think this is true,

Test Case

```
4 4
4 1
4 1
3 2
1 100
```

We solve 1,2,3 to make 1 points, Problems 1,2 don't bring us points but help increase the number of problems solved thus helping us solve problem 3 to earn 1 point.

Please let me know if I am stupid and have got it all wrong

→ Reply

depression_is_a_luxury

3 days ago, # ^ | ☆ ← Rev. 3 0

What he meant is, there exists an answer where all problems ATTEMPTED brings us points.

So, in your example, we could have just attempted the 3rd problem and it would have brought us 1 point (and 1 answer attempted only)

→ Reply

badSHAH

3 days ago, # ^ | ☆ 0

i get it, my bad. thanks for pointing out

→ Reply

badSHAH

3 days ago, # ^ | ☆ 0

please ignore the previous comment

→ Reply

GODOF_Shinobi

3 days ago, # | ☆ +5

In Problem D what if the score was only added when you have solved a[i] problems . Can anybody suggest a solution for that ?

→ Reply

muhammedaly

3 days ago, # | ☆ 0

can we solve E using Karnaugh maps??

→ Reply

kaldiuo

3 days ago, # | ☆ +5

In problem F, Strongly Connected Tournaments, can someone tell me how the law of total probability applies in each of the cases?

→ Reply

anuragsidana

2 days ago, # | ☆ 0

Why my solution failed on the 18th test case. On my machine, it's giving the right output. Why so?

34011797

→ Reply

kwangg

47 hours ago, # ^ | ☆ 0

I'm guessing it's floating-point precision errors. Probably the calculation of log_2(m) is not exact when you do log_10(m)/log_10(2) as floating-point numbers, so the comparison against n may be wrong.

→ Reply

34 hours ago, # ^ | ☆ 0

But in this test case log_2(m) is giving 25 . Hence if the

anuragsidana

But in this test case log_2(m) is giving 25 . Hence if the condition would fail in this case and the output would be m%pow(2,n) would give 0 . Where is the mistake here ?

25 33554432

→ Reply

kwangg

23 hours ago, # ^ | ☆ ← Rev. 2 0

yes, but you're storing all of these as floating-point numbers. Thus, the actual representation may not be exact, and doing equality comparisons or comparisons when the values should be equal to each other may be wrong. This might be why it failed on that test case.

It may also be some weird C++ memory thing. I also discovered that if you change both variables to float (instead of one double and one float), then it passes all testcases. See the modified submission: 34077845

→ Reply

anuragsidana

14 hours ago, # ^ | ☆ 0

Alright,thanx that's ohkay . But what if I typecast log_2(m) to long long and then compare m and log_2(m)

In that case also its not working . why so ?

→ Reply

codeRaider

2 days ago, # | ☆ +2

Has anyone solved E using Sum of Products(SOP) form, then minimizing the terms using K maps or algebraic minimization and then finally combining terms to get lexicographically smallest function ?

→ Reply

ErdemKirez

8 hours ago, # ^ | ☆ 0

I thought that it's the solution and pass the problem ASAP. However I think it can really work with some effort.

→ Reply

pk845

2 days ago, # | ☆ 0

i didn't understand problem E. can anyone please explain me that problem a little bit.

→ Reply

AlexArdelean

2 days ago, # | ☆ 0

can someone explain me please what does he consider by states in the editorial of E and what does nonterminals from the right part of the rule mean ? Help would be appreciated !

→ Reply

GyojunYoun

39 hours ago, # | ☆ ← Rev. 3 +1

I have a question about the problem F.

When we compute $cp(s, i)$, if we think that we add a new player with the largest index, then the following equation holds:

$$cp(s, i) = cp(s - 1, i) \times (1 - p)^i + cp(s - 1, i - 1) \times p^{s - i}.$$

But I thought that I add a new player with the smallest index. The equation below is what I made.

$$cp(s, i) = cp(s - 1, i) \times p^i + cp(s - 1, i - 1) \times (1 - p)^{s - i}$$

I cannot find why I was wrong. I've spent almost two days in order to solve this

problem...

Could you tell me what I am doing wrong?

→ Reply

37 hours ago, # ^ | ☆ ← Rev. 2 +1

I explain about my equation.

I'm going to add a new player whose index is the smallest. There have already been $s - 1$ players. After adding a new player, the number of players should be s.

GyojunYoun

The first term, $cp(s - 1, i) \times p^i$, describe the case when a new player is not in the set of i players who lose to the others. The 'new added player' must win all the i players who are in the set. Thus, the probability is p^i.

The second term, $cp(s - 1, i - 1) \times (1 - p)^{s - i}$, describes the case when a new player is in the set. The 'new added player' must lose to all the $s - i$ players who are not in the set. Thus, the probability is $(1 - p)^{s - i}$.

Would you tell me what is wrong?? Why should I add a largest-index-player like the official solution?

→ Reply

YakutovDmitriy

31 hour(s) ago, # ^ | ☆ +6

This way of computation of $cp(s, i)$ is correct too. Maybe you have a mistake in other part of solution.

→ Reply

GyojunYoun

31 hour(s) ago, # ^ | ☆ +1

Thank you very much! I really appreciate your help.

→ Reply

invince

28 hours ago, # | ☆ 0

In the solution of problem 913C (Party Lemonade) I didn't understand the updation of ans as `ans=min(ans,sum+(L>0)*c[i]);` Why c[i]*(L>0) is added to the sum after we have added the required money in `sum+=min(ans,sum+(L>0)*c[i]);`

→ Reply

Siriuslight

28 hours ago, # | ☆ 0

I have a binary search solution for C, I did binary search on cost available and checked if maximum volume that can be bought is greater than volume required and in this way calculated minimum price. I passed the pretests but got wrong on 14 test case, it was due to overflow. After correcting overflow I got it accepted. Here is my submission.

→ Reply

chunky_2808

13 hours ago, # | ☆ ← Rev. 3 0

In Solution of 913C ,in the following line

// ans = min(ans, sum + (L > 0) * c[i]); //

what is significance of (L>0) ????

→ Reply

Talk_less

12 hours ago, # ^ | ☆ 0

Remember that after you alter the domain as a[i] = min(2*a[i-1], a[i]), it makes sense to buy ith bottle as opposed to two i-1 bottles. So if L > 0, it means there is volume left, and it makes no sense to retain the bottles you picked previously, so get that extra ith bottle. As explained in editorial, if you set left most bit to 1, all the right side bits can be set to zero as the answer will still be maximum.

→ Reply

Reply

chunky_2808

new, 2 hours ago, # ^ | ☆ 0

No , I'm talking about code of solution.

→ Reply

ankeshgupta007

8 hours ago, # | ☆ +1

Can someone explain E clearly? I fail to understand what a state represents!

→ Reply

call_me_hopeless28

new, 115 minutes ago, # | ☆ 0

There is a recursive Greedy solution of Problem C. Watch submission --> 34095086

→ Reply

Codeforces (c) Copyright 2010-2017 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Jan/12/2018 02:40:32$^{UTC+5.5}$ (c5).
Desktop version, switch to mobile version.
Privacy Policy