1. Introduction to the judge enviroment

2. Partial sums/precomputation trick

Sample Problem: Even Pairs

Input:

- first line: a positive integer n
- second line: a sequence $x_0, \ldots, x_{n-1} \in \{0, 1\}$

Sample Problem: Even Pairs

Input:

- first line: a positive integer n
- second line: a sequence $x_0, \ldots, x_{n-1} \in \{0, 1\}$

Output: a single line containing the number of pairs $0 \leq i \leq j < n$ such that

$$x_i + \cdots + x_j$$

is even.

First approach

(1) for all pairs $i \leq j$, compute the sum $x_i + \cdots + x_j$

(2) if it is even, increment a counting variable

First approach

(1) for all pairs $i \leq j$, compute the sum $x_i + \cdots + x_j$

(2) if it is even, increment a counting variable

A few points, but also a timelimit error on harder test cases!

- this means that our algorithm is too slow

First approach

(1) for all pairs $i \leq j$, compute the sum $x_i + \cdots + x_j$
(2) if it is even, increment a counting variable

A few points, but also a timelimit error on harder test cases!

- this means that our algorithm is too slow
- we have three nested loops: two for going over all pairs $i \leq j$, and one for summing up the $x_i + \cdots + x_j$

First approach

(1) for all pairs $i \leq j$, compute the sum $x_i + \cdots + x_j$

(2) if it is even, increment a counting variable

A few points, but also a timelimit error on harder test cases!

- this means that our algorithm is too slow
- we have three nested loops: two for going over all pairs $i \leq j$, and one for summing up the $x_i + \cdots + x_j$
- running time is $\Theta(n^3)$.
- this type of analysis is very important in this course.

Second approach

Observation: if we know the parity of the sum

$$x_i + \cdots + x_j$$

then based on the parity of x_{j+1} we also know the parity of

$$x_i + \cdots x_j + x_{j+1}$$

Second approach

Observation: if we know the parity of the sum

$$x_i + \cdots + x_j$$

then based on the parity of x_{j+1} we also know the parity of

$$x_i + \cdots x_j + x_{j+1}$$

Running time: $\Theta(n^2)$

Third approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=1}^{j} x_a - \sum_{b=1}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

Third approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=1}^{j} x_a - \sum_{b=1}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=1}^{i} x_a$ in one iteration

(2) for every $i \leq j$ check the parity of $S_j - S_{i-1}$

Third approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=1}^{j} x_a - \sum_{b=1}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=1}^{i} x_a$ in one iteration

(2) for every $i \leq j$ check the parity of $S_j - S_{i-1}$

Running time: $\Theta(n^2)$

Fourth approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=1}^{j} x_a - \sum_{b=1}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=1}^{i} x_a$ in one iteration
(2) for every j: $E_j = \#$ of S_i $(i \leq j)$ that are even
(3) for every j: $O_j = \#$ of S_i $(i \leq j)$ that are odd

Fourth approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=1}^{j} x_a - \sum_{b=1}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=1}^{i} x_a$ in one iteration
(2) for every j: $E_j = \#$ of S_i $(i \leq j)$ that are even
(3) for every j: $O_j = \#$ of S_i $(i \leq j)$ that are odd
(4) if S_j is even: increase the counter by $E_{j-1} + 1$
    if S_j is odd: increase the counter by O_{j-1}

Fourth approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=1}^{j} x_a - \sum_{b=1}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=1}^{i} x_a$ in one iteration
(2) for every j: $E_j = \#$ of S_i $(i \leq j)$ that are even
(3) for every j: $O_j = \#$ of S_i $(i \leq j)$ that are odd
(4) if S_j is even: increase the counter by $E_{j-1} + 1$
    if S_j is odd: increase the counter by O_{j-1}

Running time: $\Theta(n)$

Even pairs - conclusion

Tricks/techniques: Partial sums/Precomputing

- Precomputing partial sums allows computing the sum of the elements in an interval in constant time.
- More generally, precomputing certain values can speed up the running time of an algorithm.

Judge Results

Besides correct, timelimit and wrong-answer, the judge can give the following results.

assertion-failure	SIGABRT: memory screwup or assertion failure
segmentation-fault	SIGSEGV: memory screwup
run-error	nonzero exit status
forbidden	bad syscall or other safety

Forum

The forum is your main tool for discussing ideas and getting help. Use it.

Of course, you will only learn if you first try to solve the problems **on your own**.

Forum: How To Ask Questions

1. Apply spoiler warnings

Example

SPOILER<<<
Set this text to have a white foreground. It will then be invisible unless marked. The <<<...>>> exploit a bug in the email plugin to also remove the text in plain-text email.
>>>

Forum: How To Ask Questions

1. Apply spoiler warnings
2. Describe the problem, not your guesses or summaries

Example

Bad When I compile, it tells me it cannot find it.

Good When I run `g++ -o foo foo.cpp`, I get

```
bash: $'g++\302\240-o': command not found
```

Forum: How To Ask Questions

1. Apply spoiler warnings
2. Describe the problem, not your guesses or summaries
3. Code: describe what fails and what you expect instead

Example

Bad The code below doesn't work. Help?

Good I am trying to solve problem 1. I tried strategy blah blah. My code is below. For some reason, when running it on the provided testcase it emits `no solution` instead of `1`. What am I doing wrong?

Forum: How To Ask Questions

1. Apply spoiler warnings
2. Describe the problem, not your guesses or summaries
3. Code: describe what fails and what you expect instead
4. Code: post minimal examples

Example

Bad When I call `.foo()` on a `vector`, it segfaults. Bug!

Good I am trying to blah. The code is below. I get a segfault in the line that calls `.foo()`, but if I remove that line the program continues. What am I doing wrong?

Forum: How To Ask Questions

1. Apply spoiler warnings
2. Describe the problem, not your guesses or summaries
3. Code: describe what fails and what you expect instead
4. Code: post minimal examples
5. Don't rush to claim that you have found a bug

Further reading: How To Ask Questions The Smart Way

C++

Learning C++ is beyond the scope of this course.

- True beginners should probably read a book
  - Andrew Koenig and Barbara E. Moo: Accelerated C++, Addison-Wesley, 2000
  - Stanley B. Lippman: C++ Primer, 3rd ed., Addison-Wesley, 1998

C++

Learning C++ is beyond the scope of this course.

- True beginners should probably read a book
  - Andrew Koenig and Barbara E. Moo: Accelerated C++, Addison-Wesley, 2000
  - Stanley B. Lippman: C++ Primer, 3rd ed., Addison-Wesley, 1998
- People familiar with the syntactic family (e.g. C, Java, C#) may get away with a tutorial

C++

Learning C++ is beyond the scope of this course.

- True beginners should probably read a book
  - Andrew Koenig and Barbara E. Moo: Accelerated C++, Addison-Wesley, 2000
  - Stanley B. Lippman: C++ Primer, 3rd ed., Addison-Wesley, 1998
- People familiar with the syntactic family (e.g. C, Java, C#) may get away with a tutorial
- Useful in any case: C++ FAQ Lite (link)

C++

STL is part of the C++ standard library. It is important that you know

- how and when to use the classes `std::vector`, `std::priority_queue`, `std::set` and `std::map`,
- how to do I/O using `<iostream>`, and
- how to use the `sort` function from the `<algorithm>` header file.

That's it.

The most important things to remember:

- You can find all that you need at our website: `http://www.cadmo.ethz.ch/education/lectures/HS16/algolab/index.html`

That's it.

The most important things to remember:

- You can find all that you need at our website: `http://www.cadmo.ethz.ch/education/lectures/HS16/algolab/index.html`
- For questions, you should use the forum.

That's it.

The most important things to remember:

- You can find all that you need at our website: `http://www.cadmo.ethz.ch/education/lectures/HS16/algolab/index.html`
- For questions, you should use the forum.
- Today, we will publish the first week's exercises on the webpage.

That's it.

The most important things to remember:

- You can find all that you need at our website: `http://www.cadmo.ethz.ch/education/lectures/HS16/algolab/index.html`
- For questions, you should use the forum.
- Today, we will publish the first week's exercises on the webpage.
- Next Monday at 17:00, we have the first problem of the week.