1. Introduction to the judge environment

2. Prefix sum/precomputation technique

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 sets!

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 sets!

- 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 sets!

- 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 sequence $x_i + \cdots + x_j$

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 sets!

- 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 sequence $x_i + \cdots + x_j$
- running time is $\Theta(n^3)$

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 sets!

- 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 sequence $x_i + \cdots + x_j$
- running time is $\Theta(n^3)$
- this type of analysis is **very important** in this course.

Second Approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=0}^{j} x_a - \sum_{b=0}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

Second Approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=0}^{j} x_a - \sum_{b=0}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=0}^{i} x_a$ in one iteration

Second Approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=0}^{j} x_a - \sum_{b=0}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=0}^{i} x_a$ in one iteration

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

Second Approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=0}^{j} x_a - \sum_{b=0}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=0}^{i} x_a$ in one iteration

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

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

Third Approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=0}^{j} x_a - \sum_{b=0}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=0}^{i} x_a$ in one iteration

Third Approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=0}^{j} x_a - \sum_{b=0}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=0}^{i} x_a$ in one iteration

(2) $E = \#$ of S_i that are even

(3) $O = \#$ of S_i that are odd

Third Approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=0}^{j} x_a - \sum_{b=0}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=0}^{i} x_a$ in one iteration

(2) $E = \#$ of S_i that are even

(3) $O = \#$ of S_i that are odd

(4) result is $\binom{E}{2} + \binom{O}{2} + E$

Third Approach

Observation:

$$x_i + \cdots + x_j = \sum_{a=0}^{j} x_a - \sum_{b=0}^{i-1} x_b$$
$$= S_j - S_{i-1}$$

(1) calculate partial sums $S_i = \sum_{a=0}^{i} x_a$ in one iteration

(2) $E =$ # of S_i that are even

(3) $O =$ # of S_i that are odd

(4) result is $\binom{E}{2} + \binom{O}{2} + E$

Running time: $\Theta(n)$

Even Pairs - Conclusion

Technique: 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 Feedback

Besides CORRECT, TIMELIMIT, and WRONG-ANSWER, the judge can give the following feedback:

ASSERTION-FAILURE SIGABRT: memory screw-up or assertion failure

SEGMENTATION-FAULT SIGSEGV: memory screw-up (e.g. out-of-bounds)

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 **something**. My code is below. For some reason, when running it on the provided test case 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 **something**. 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

C++

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

- True beginners should probably read a book

C++

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

- True beginners should probably read a book
- 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
- People familiar with the syntactic family (e.g. C, Java, C#) may get away with a tutorial
- Useful in any case: C++ FAQ Lite

C++

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

- True beginners should probably read a book
- People familiar with the syntactic family (e.g. C, Java, C#) may get away with a tutorial
- Useful in any case: C++ FAQ Lite
- **NEW!** Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

C++

NEW! Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

C++

NEW! Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

Contains:

- Easy exercises to get you started with C++ including Judge feedback

C++

NEW! Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

Contains:

- Easy exercises to get you started with C++ including Judge feedback
- Handling I/O

C++

NEW! Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

Contains:

- Easy exercises to get you started with C++ including Judge feedback
- Handling I/O
- Fundamental data types and basic concepts of the C++ standard library

C++

NEW! Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

Contains:

- Easy exercises to get you started with C++ including Judge feedback
- Handling I/O
- Fundamental data types and basic concepts of the C++ standard library
- How to compile, run, and debug

C++

NEW! Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

Contains:

- Easy exercises to get you started with C++ including Judge feedback
- Handling I/O
- Fundamental data types and basic concepts of the C++ standard library
- How to compile, run, and debug

Does not contain:

C++

NEW! Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

Contains:

- Easy exercises to get you started with C++ including Judge feedback
- Handling I/O
- Fundamental data types and basic concepts of the C++ standard library
- How to compile, run, and debug

Does not contain:

- Introduction to C++ programming

C++

NEW! Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

Contains:

- Easy exercises to get you started with C++ including Judge feedback
- Handling I/O
- Fundamental data types and basic concepts of the C++ standard library
- How to compile, run, and debug

Does not contain:

- Introduction to C++ programming
- Control structures

C++

NEW! Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

Contains:

- Easy exercises to get you started with C++ including Judge feedback
- Handling I/O
- Fundamental data types and basic concepts of the C++ standard library
- How to compile, run, and debug

Does not contain:

- Introduction to C++ programming
- Control structures
- Memory management

C++

NEW! Updated version of the ‘A Short Introduction to C++ for the Algorithms Lab’

Contains:

- Easy exercises to get you started with C++ including Judge feedback
- Handling I/O
- Fundamental data types and basic concepts of the C++ standard library
- How to compile, run, and debug

Does not contain:

- Introduction to C++ programming
- Control structures
- Memory management
- Objected Oriented Programming in C++

C++

STL (Standard Template Library) is part of the C++ standard library. It is important that you know:

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

C++

STL (Standard Template Library) is part of the C++ standard library. It is important that you know:

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

IMPORTANT! Read the 'Introduction to C++' document and familiarise yourselves with the concepts within.

That's it!

The most important things to remember:

- You can find all that you need at our website:
  https://www.cadmo.ethz.ch/education/lectures/HS18/algolab/index.html

That's it!

The most important things to remember:

- You can find all that you need at our website:
  https://www.cadmo.ethz.ch/education/lectures/HS18/algolab/index.html
- For questions, you should use the forum:
  https://moodle-app2.let.ethz.ch/course/view.php?id=4690

That's it!

The most important things to remember:

- You can find all that you need at our website:
  https://www.cadmo.ethz.ch/education/lectures/HS18/algolab/index.html
- For questions, you should use the forum:
  https://moodle-app2.let.ethz.ch/course/view.php?id=4690
- Today, we will publish the first week's exercises on the Moodle.

That's it!

The most important things to remember:

- You can find all that you need at our website:
  `https://www.cadmo.ethz.ch/education/lectures/HS18/algolab/index.html`
- For questions, you should use the forum:
  `https://moodle-app2.let.ethz.ch/course/view.php?id=4690`
- Today, we will publish the first week's exercises on the Moodle.
- The public input/output files are available in the zip file Week 1 Zipfile on Moodle.

That's it!

The most important things to remember:

- You can find all that you need at our website:
  `https://www.cadmo.ethz.ch/education/lectures/HS18/algolab/index.html`
- For questions, you should use the forum:
  `https://moodle-app2.let.ethz.ch/course/view.php?id=4690`
- Today, we will publish the first week's exercises on the Moodle.
- The public input/output files are available in the zip file Week 1 Zipfile on Moodle.
- Today, we will also publish easy exercises to help you get started with C++.

That's it!

The most important things to remember:

- You can find all that you need at our website:
  `https://www.cadmo.ethz.ch/education/lectures/HS18/algolab/index.html`
- For questions, you should use the forum:
  `https://moodle-app2.let.ethz.ch/course/view.php?id=4690`
- Today, we will publish the first week's exercises on the Moodle.
- The public input/output files are available in the zip file Week 1 Zipfile on Moodle.
- Today, we will also publish easy exercises to help you get started with C++.
- Next Monday at 17:00, we have the first Problem of The Week!