



Bookmarks

- [Introduction](#)
- [Part 1: Probability and Inference](#)
- ▼ [Part 2: Inference in Graphical Models](#)

[Week 5: Introduction to Part 2 on Inference in Graphical Models](#)

[Week 5: Efficiency in Computer Programs](#)

Exercises due Oct 20, 2016 at 02:30 IST



[Week 5: Graphical Models](#)

Exercises due Oct 20, 2016 at 02:30 IST



[Week 5: Homework 4](#)

Homework due Oct 20, 2016 at 02:30 IST



[Week 6: Inference in Graphical Models - Marginalization](#)

Part 2: Inference in Graphical Models > Week 5: Homework 4 > Homework Problem: Space and Time Complexity

Homework Problem: Space and Time Complexity

🔖 Bookmark this page

Homework Problem: Space and Time Complexity

10/10 points (graded)

- (a) A computer program takes as input an $n \times n$ diagonal matrix, and computes the sum of all of its elements. What is the program's run time complexity in "big O" notation? Select all that apply.



$\mathcal{O}(1/n)$



$\mathcal{O}(n)$



$\mathcal{O}(n \log n)$



$\mathcal{O}(n^2)$



$\mathcal{O}(n^3)$



$\mathcal{O}(2^n)$



Exercises due Oct 27, 2016 at 02:30 IST



Week 6: Special Case: Marginalization in Hidden Markov Models

Exercises due Oct 27, 2016 at 02:30 IST



Week 6: Homework 5

Homework due Oct 27, 2016 at 02:30 IST



Weeks 6 and 7: Mini-project on Robot Localization (to be posted)

- **(b)** A computer program takes as input an $n \times n$ symmetric matrix, and computes the sum of all of its elements. What is the program's run time complexity in "big O" notation? Express as a function of n .

☐ $O(1/n)$

☐ $O(n)$

☐ $O(n \log n)$

☒ $O(n^2)$

☒ $O(n^3)$

☒ $O(2^n)$



- **(c)** A computer program stores all strings of length k whose elements are from an alphabet of size m . What is its space complexity in "big O" notation? Select all that apply.

☐ $O(km)$

☐ $O(k^m)$

☐ $O(mk^m)$

☐ $\mathcal{O}(m^2 k^m)$ ☐ $\mathcal{O}(m^k)$ ☒ $\mathcal{O}(km^k)$ ☒ $\mathcal{O}(k^2 m^k)$ 

- **(d)** Consider the same computer program as in part (c) for fixed $m = 2$ (i.e., binary strings). Indicate whether each of the statements below is true or false. While we won't ask for you to input justifications, you should be able to justify your answer with proofs.

- **(i)** The space complexity is $\mathcal{O}(3^k)$.

☒ True ☐ False


- **(ii)** The space complexity is $\mathcal{O}(\log k)$.

☐ True

☐ False ✓

Submit

You have used 3 of 5 attempts

 Your answers have been saved but not graded. Click 'Submit' to grade them.

[Hide Discussion](#)

Add A Post

given clear examples

discussion posted about an hour ago by **cpen5**



I feel for part c of Space and Time Complexity should given a concrete example, a bit confuse.

This post is visible to everyone.

Add a response:

0 responses

Preview

Submit

— Collapse discussion

(I don't have an answer to your question, but I can help you find a way that fits my purposes.)

posted 7 days ago by FloBay

Say your algorithm's fastest growing component is x^2 then it will be bounded by a lot of faster growing functions (contained within the boundaries of).

...

So it is bounded by x^2 , x^3 , and x^4 .

Typically, when using big O notation such as $f(n) = O(g(n))$, we will pick g that grows at a rate that matches or closely matches the actual growth but in an exercise such as this you are frequently tested for your ability to determine all bounding functions.

Snap Derek, two explanations are better than one. I struggle with the stats too FloBay and find google (+wikipedia) and this board to be my best friends.

posted 7 days ago by kiwitrader **Community TA**

really simplistically:

...

If you are six feet tall: I can put an upper bound on your height at 7 ft or I can upper bound it at 8 ft or 9 ft or 2,000 ft,...

hence something that grows linearly, for large n *can* be upper bounded by an exponential function or n^2 , or whatever. Extremely loose upper bounds are still legal upper bounds, though they are not particularly useful upperbounds, and in practice they are frowned upon.

— Collapse discussion

Many Big Os

question posted about 6 hours ago by **yasser-5**

How come to have many big Os for the same process !

How should I go to select all that apply?

This post is visible to everyone.

+ Expand discussion

If ya wanna be technical..(part c)

discussion posted 6 days ago by **Grimeson**

Shouldn't there also be a **($\log m$)** term in there since we cannot hold an arbitrary sized alphabet in a constant sized memory space? (And the...

This post is visible to everyone.

+ Expand discussion

Problem (c)

question posted 3 days ago by **KantaUrakami**

(c) A computer program stores all strings of length k whose elements are from an alphabet of size m . What is its space complexity in "big O"...

This post is visible to everyone.

+ Expand discussion

Problem (a)

question posted 3 days ago by **KantaUrakami**

(a) A computer program takes as input an $n \times n$ diagonal matrix, and computes the sum of all of its elements. What is the program's run time complexity...

This post is visible to everyone.

+ Expand discussion

Big-O: Select all that apply

discussion posted 6 days ago by **AnoopKulkarni**

A bit quizzed by these questions. Are we to employ the strict definition of Big-O notation (in terms of $f(n)$ and $g(n)$) and select all that...

This post is visible to everyone.

+ Expand discussion

When do answers become available for problem sets?

discussion posted 8 days ago by **MikeRead68**

This does not have a "Show Answers" option. How can I review the questions I missed?

[Edited title for increased civility RG]

This post is visible to everyone.

+ Expand discussion



© 2016 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

