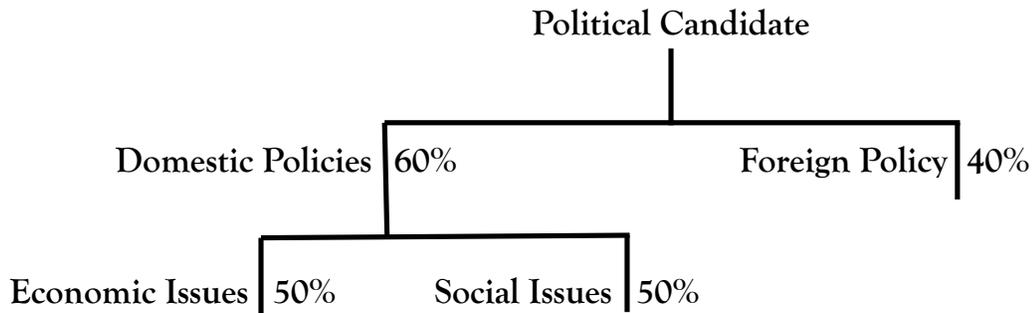


SIXTH QUIZ

You have 15 minutes from the start of class to complete this quiz. Give partial answers if you can't give complete ones. Read the questions with care; work with deliberate speed. Don't give us more than we ask for. The usual instructions apply. Good luck!

Problem 1 (8 points)

Below is a relevance tree for political candidates. Fill in the five blanks to calculate the ratings of each alternative. Showing your work might increase your chances of getting partial credit if your arithmetic is wrong.



Overall relevance
of each node:

Laurie Legislator	5	6	10	_____
Samuel Solon	10	8	2	_____

Ratings of
alternatives

Problem 2 (4 points)

In class we discussed formal verification and n-version programming as solutions proposed to the “software crisis” of expensive, unreliable software. For each of the two techniques, briefly give one major reason it didn't solve the software crisis (other than “it's too time-consuming and expensive”).

- (a) Formal verification didn't solve the software crisis because

- (b) N-version programming didn't solve the software crisis because

Problem 3 (5 points)

Suppose you have a 50% chance of getting a summer job at Google, a 25% chance of working at Microsoft, and a 25% chance of working at Apple. Suppose that Google and Microsoft would each pay you \$10,000 and Apple would pay you \$12,000. What's your expected salary this summer (i.e., what's the expected value)?

Problem 4 (3 points)

(a) What was the O -notation of the fastest sorts shown in the film *Sorting Out Sorting*?

(b) What was the O -notation of the slowest sorts shown in the film *Sorting Out Sorting*?

(c) *Sorting Out Sorting* showed three categories of sorting algorithms: exchange sorts, insertion sorts, and selection sorts. Did any of these categories guarantee you that your algorithm wouldn't be in the slowest O -notation group?