Homework (4)

Deadline: 1404 / 03 / 18

English for Computing

Spring 1404

Tips

1. Using GPT is not allowed for answering the questions.
2. With each day of delay in submitting the answer sheet, you will lose 25% of your homework score.
3. Name your answer sheet using the format “ESP-SID-Name”.
4. Please upload your answer sheet in the appropriate section of the course page in CW.
5. Make sure you have written your name and student ID in your answer sheet.

Question 1

How Video Travels Across the Internet

As any end-user can tell you, watching streaming video online is rarely seamless. Streams suffer from hiccups, drop-outs, and sudden connection loss. This section explains how the Internet moves data and why these characteristics affect video playback.

It can be puzzling that the Internet struggles with audio and video when radio, television, and telephones have carried them smoothly for decades. Let’s look first at those traditional media.

Radio: A single tower broadcasts one signal to many receivers; everyone hears the same programme at the same time simply by tuning to the right frequency. Communication is one-way.

Television: TV works like radio but on a national scale: the same programme is relayed through networks of towers, cable companies, or satellites to viewers across the country – again, a one-way broadcast.

Telephone: Voice calls reuse many of the same physical wires as the Internet, but a telephone switch reserves a dedicated two-way circuit for the duration of the call – a technique known as circuit switching.

Internet Basics: A web session is also two-way – the browser requests, the server responds – yet no circuit is reserved. Instead, the data are chopped into packets that travel independently across a series of routers. Each boundary between Internet Service Providers (ISPs) or network types is crossed by hopping from router to router, a process called packet switching.

Packet switching delivers robustness – lost packets can be resent – but also unpredictability. Because packets often follow different routes, some arrive quickly, others slowly, and a few never arrive at all. Every extra “hop” adds latency (delay). Video is therefore prone to pauses, stutters, and skips, whereas small web pages tolerate these delays far better.

Finally, overall throughput is limited by the narrowest link in the chain – your video can only stream as fast as the slowest segment of the path, a characteristic commonly (but imprecisely) labelled bandwidth.

Section A: Vocabulary in Context

Define each term in your own words and use it in an original sentence related to computer networking:

- ➔ Broadcast
- ➔ Circuit switching
- ➔ Packet
- ➔ Router
- ➔ Latency
- ➔ Bandwidth
- ➔ Internet backbone

Section B: Reading Comprehension

Answer the following questions based on the passage:

1. Why does radio transmission avoid the problems seen with Internet video streams?
2. In what fundamental way does circuit switching differ from packet switching?
3. According to the text, what role do routers play when a packet crosses from one ISP to another?

4. Give two reasons why packets of the same video file might arrive at very different times.
5. Explain in one sentence why a “fast” hosting provider cannot guarantee smooth video to every viewer.

Section C: True / False / Not Given

Indicate whether each statement is True (T), False (F), or Not Given (NG) based on the passage:

1. Radio and television are examples of one-way communication.
2. Telephone circuits release unused bandwidth whenever the caller is silent.
3. Packet loss never occurs on the public Internet thanks to retransmission mechanisms.
4. Each additional network hop increases the likelihood of variable latency.
5. The passage claims that raising bandwidth always eliminates video buffering.

Section D: Short Writing (120 – 150 words)

Summarize how packet switching both helps and hinders streaming video, and propose **one** practical strategy a content provider could use to mitigate the negative effects.

Question 2

Protein-Protein Interaction (PPI) Prediction Methods

The rapid growth of high-throughput experimental methods has generated a large volume of data on protein–protein interactions (PPIs). However, these experimental datasets often contain significant noise, including both false positives and false negatives. For example, techniques such as mass spectrometry may fail to detect transient or weak interactions. Overall, various high-throughput technologies—including two-hybrid systems, mass spectrometry, protein chips, and phage display—are associated with relatively high noise levels.

Studying PPIs is foundational for diagnostic and therapeutic applications, enabling the design of novel drugs. To overcome the limitations of experimental methods, computational approaches have been developed to predict PPIs more efficiently and accurately.

Recent advances in computational modeling have led to important achievements in protein design, including enzyme engineering, development of new therapies, biosensor creation, and discovery of small-molecule binders. While these approaches excel at modifying naturally occurring proteins, creating proteins de novo offers complete control over protein structure and function. This shift allows the discovery of novel folds or structural elements as building blocks for innovative proteins.

Computational protein design aims to automate the creation of proteins with specific structural and functional properties. Over the past two decades, this field has advanced significantly, enabling the design of new three-dimensional folds, protein complexes, and enzymes. Despite these successes, current methods remain somewhat unreliable; initial designs often require multiple cycles of trial and error due to the complexities of accurately modeling protein physics and sampling protein sequence space.

Nevertheless, computational approaches have facilitated the generation of synthetic protein domains that mimic natural folds using novel amino acid sequences.

Efficient computational screening of many possible protein designs can significantly reduce experimental workload and save time. Computational PPI prediction methods vary: some utilize extracted protein features as input for learning models, while others focus on deriving new protein information from structural data. Experimental techniques such as X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy provide detailed three-dimensional structures of proteins, which can help identify interaction sites. However, these experimental methods are labor-intensive and time-consuming, limiting the availability of structural data for many proteins.

To address this gap, deep learning methods have emerged to predict potential PPIs based on protein structural and sequence features. These computational tools help infer interaction sites and enhance understanding of protein function.

PPI Site Prediction

Identifying PPI binding sites is critical for elucidating disease mechanisms and for designing novel therapeutics. PPI binding sites consist of amino acid residues that form chemical bonds with other molecules. Determining these interaction domains aids in understanding cellular regulatory processes, locating drug targets, and predicting protein functions.

PPI site prediction methods generally fall into three categories: protein–protein docking, structure-based approaches, and sequence-based approaches. Docking methods aim to generate models of protein complexes by evaluating shape complementarity and scoring potential binding configurations.

Recent sequence-based approaches employ advanced machine learning techniques, such as attention-based convolutional neural networks, simplified long short-term memory (LSTM) models, and hybrid models combining convolutional and recurrent neural networks. These methods integrate local contextual information and global sequence features to improve prediction accuracy and address challenges like data imbalance.

In summary, computational prediction of protein–protein interactions and their binding sites is a rapidly evolving field that complements experimental approaches. Advances in machine learning and structural bioinformatics have significantly enhanced our ability to identify interaction sites and characterize PPIs with greater accuracy. These developments are crucial for accelerating drug discovery and deepening our understanding of complex biological processes. Continued integration of computational predictions with experimental validation promises to unlock new opportunities for therapeutic intervention and the design of novel proteins with tailored functions.

Section A: Short-Answer Questions

1. What structural feature of PPI interfaces makes designing small-molecule modulators challenging?
2. Name two experimental methods commonly used to determine protein tertiary structures.
3. What is the main advantage of fragment-based drug discovery over high-throughput screening?
4. What computational approach uses protein tertiary structures to predict potential protein–protein interaction sites?
5. Why are aberrant PPIs considered promising targets for novel drug therapies?

Section B: Long-Answer / Discussion Questions

(Limit your answer to a maximum of 3 lines for each question.)

1. Explain why PPI interfaces tend to be "undruggable" compared to traditional protein targets.
2. Discuss the role of machine learning in improving the prediction accuracy of PPI binding sites.
3. Compare the challenges and benefits of de novo protein design versus modifying naturally occurring proteins.

Section C: Fill-in-the-Blank Items

1. PPI interfaces are often ________ and lack deep binding pockets, making small-molecule targeting difficult.
2. ________ and ________ are two common experimental techniques used to determine the 3D structure of proteins.
3. Fragment-based drug discovery (FBDD) typically identifies low-affinity ________ which can be optimized into effective modulators.
4. ________ screening relies on computational models to identify potential drug candidates from large compound libraries.
5. The smallest functional unit in a protein sequence that interacts with another protein is called a ________.
6. The smallest functional unit in a protein sequence that interacts with another protein is called a ________.

Section D: Multiple-Choice and True–False

1. Multiple Choice – Which of the following computational techniques predicts PPI sites by modeling proteins as graphs?
   A. Molecular Dynamics
   B. Graph Neural Networks
   C. Fourier Transform
   D. Genetic Algorithms

2. Multiple Choice – Approximately how many protein–protein interactions are estimated to exist in the human interactome?
   A. 1,000–5,000
   B. 10,000–50,000
   C. 130,000–650,000
   D. Over 1 million

3. True or False: High-throughput experimental methods for PPI detection generally produce completely accurate and noise-free datasets.

4. Multiple Choice – Which factor does NOT typically hinder small molecule binding to PPI interfaces?
   A. Large interaction surface area
   B. Presence of hydrophobic patches
   C. Highly flexible binding pockets
   D. Lack of well-defined grooves or pockets

Question 3

PASSAGE 1

The programming process can be subdivided into a number of tasks of which debugging is perhaps one of the most pervasive. For example, it is frequently estimated that testing and debugging account for approximately 50% of the cost of new systems development. This situation holds also for the later maintenance phase. Furthermore, the introduction of new approaches such as fourth-generation languages does not immediately threaten much of the current mode of system development and enhancement.

Debugging can be regarded as a problem-solving task in which the interaction between programmers' knowledge structures and environmental variables determines the effectiveness of debugging performance. While acknowledging that individual differences play a major role in debugging effectiveness, this paper investigates one of the environmental variables--the nature of the program bug--to examine its effect on the debugging process. The ultimate aim of the paper is to contribute to a general theory of program bugs.

Debugging studies such as those by Gould & Drongowski (1974) and Gould (1975) addressed the effect of the type of programming language statement containing the bug on debugging performance. They found that assignment bugs were much harder to detect and correct than were array or iteration bugs.Unfortunately, the routines Gould used were statistical routines and the assignment bugs involved knowledge of statistics, 1.e. domain knowledge not commonly possessed by the average programmer. These results highlight the anomalies that may arise when we do not have a general concept of program bugs. Furthermore, they illustrate the need to investigate the nature of bugs if research into debugging is to progress.

Questions

1. The word “pervasive” in paragraph 1 Is closest in meaning to
   A. prevalent
   B. difficult
   C. perilous
   D. technical

2. The word “they” in paragraph 3 refers to ....
   A. the anomalies
   B. program bugs
   C. these results
   D. statistical routines

3. The passage is probably taken from which of the following parts of an article?
   A. Discussion
   B. Introduction
   C. Abstract
   D. Methodology

4. According to the passage, which of the following statements is true?
   A. Testing alone accounts for 25% of the cost of new systems development.
   B. Individual differences almost play no role in the effectiveness of debugging.
   C. Based on the debugging studies, array or iteration bugs were less difficult to identify than assignment bugs.
   D. Debugging studies such as those by Gould & Drongowski were published in the second half of the 19th century.

5. The passage provides sufficient information to answer which of the following Questions?

   I. What is the most frequent task in the programming process?

   II. Who first used the term bug in the context of computer programming?

   III. What is the prerequisite for the progress of research into debugging?

   1) Only I

   2) Only II

   3) I and III

   4) Only III

PASSAGE 2

In parallel with the growing recognition of computer design and engineering as a distinct area of professional activity, computer development moved from universities to the private sector. As has been well documented, many companies entered the field in the 1950s. In fact, individuals from companies rather than universities dominated the organizing committee of the first JCC. This new industry was also where the identity of the computer engineer was ever more solidly linked to preexisting academic credentials (such as degrees in electrical engineering), specialized expertise, control over specific work tasks, and differentiation from other professionals. The case of Engineering Research Associates (ERA) offers further insights about the position of engineers in one pioneering firm. Founded in the mid-1940s, this Minneapolis-based company's early activities included developing electronic data-processing systems, especially for the US Navy; it was later well known for its stored-program computer, the 1101.

The company's founders and early personnel included a roughly even mix of mathematicians, electrical engineers, and physicists. Yet as the firm grew, engineers increasingly filled its ranks. Per Arthur Norberg, "40 percent of the 1943 electrical engineering graduates of the University of Minnesota . . . joined ERA after the war and a significant number . . . of the class of 1951 accepted their first job at ERA." Additional details can be gleaned from a 1952 personnel directory, which reveals that more than 60 percent of the company's "professional" staff held

engineering degrees. Furthermore, more than 40 percent of this same group had at least one degree in electrical engineering, and between half and two-thirds of the company's directors and vice presidents held electrical engineering degrees.

Questions

1. The passage mentions all of the following terms EXCEPT ....
    A. human resource
    B. personnel directory
    C. specialized expertise
    D. academic credentials

2. Why does the author mention ERA in paragraph 1?
    A. As it was famous for its high-tech laboratories
    B. As it was a company where theory and practice best intersected
    C. As it demonstrates the position of engineers in a leading company
    D. As it was the most advanced engineering firm cooperating with the US Navy

3. According to paragraph 2, ERA's founders and early personnel included all of the following EXCEPT ....
    A. physicists
    B. mathematicians
    C. electrical engineers
    D. electronics engineers

4. According to the passage, which of the following statements is true?

A. People from universities rather than companies made up the majority of the organizing committee of the first JCC.

B. Developing electronic data-processing systems was among the early activities of Engineering Research Associates.

C. More than half of the 1943 electrical engineering graduates of the University of Minnesota joined ERA after the war.

D. As the recognition of computer design and engineering as a separate area increased, computer development moved from the industry to academia.

5. The passage provides sufficient information to answer which of the following questions?

I. Where was the computer first invented?

II. In which decade was ERA established?

III. How many of the US companies' staff held electrical engineering degrees?

A. Only I

B. Only II

C. Only III

D. II and III

Good luck!
English for Computing education team