CSCI2100C 2019-20: Solution 1[*]

This assignment is due at 11:59:59pm, 7th March 2019.

- **Q1. [10 marks]** Consider the algorithm minSubArraySum shown below, what is the asymptotic (**Big-Oh**) complexity of this algorithm in the worst case?

```
Algorithm minSubArraySum(A, n)
1:  minSum=A[0]
2:  for i = 0 to n − 1
3:      subSumFromI=0
4:      for j = i to n − 1
5:          subSumFromI+=A[j]
6:              if subSumFromI<minSum
7:                  minSum=subSumFromI
8:  return minSum
```

1: $O(1)$
2: $O(n)$
3: $O(n)$
4: $O(n^2)$
5: $O(n^2)$
6: $O(n^2)$
7: $O(n^2)$
8: $O(1)$
$O(\max(1, n, n, n^2, n^2, n^2, n^2, 1)) = O(n^2)$

- **Q2. [26 marks]** Answer the following questions related to $O(\cdot)$, $\Omega(\cdot)$ and $\Theta(\cdot)$.
  - **(i). [8 marks]** Prove the correctness of the product property of $O(\cdot)$. Specifically, prove: if $g_1(n) = O(f_1(n))$ and $g_2(n) = O(f_2(n))$, then $g_1(n) \cdot g_2(n) = O(f_1(n) \cdot f_2(n))$. (Hint. Using the definition. If $g_i(n) = O(f_i(n))$, then there exists constants c_i and n_i such that $g_i(n) \le c_i \cdot f_i(n)$ for $n \ge n_i$.)

[*]Departmental Guideline for Plagiarism (Department of Systems Engineering and Engineering Management): If a student is found plagiarizing, his/her case will be reported to the Department Examination Panel. If the case is proven after deliberation, the student will automatically fail the course in which he/she committed plagiarism. The definition of plagiarism includes copying of the whole or parts of written assignments, programming exercises, reports, quiz papers, mid-term examinations and final examinations. The penalty will apply to both the one who copies the work and the one whose work is being copied, unless the latter can prove his/her work has been copied unwittingly. Furthermore, inclusion of others' works or results without citation in assignments and reports is also regarded as plagiarism with similar penalty to the offender. A student caught plagiarizing during tests or examinations will be reported to the Faculty office and appropriate disciplinary authorities for further action, in addition to failing the course.

According to the definition of O-notation, there exist c_1, n_1 and c_2, n_2 when $n \ge n_1$, $n \ge n_2$, $g_1(n) \le c_1 \cdot f_1(n)$ and $g_2(n) \le c_2 \cdot f_2(n)$ respectively. For $n \ge n_0 = \max(n_1, n_2)$, $g_1(n) \le c_1 \cdot f_1(n)$, $g_2(n) \le c_2 \cdot f_2(n)$ so that $g_1(n) \cdot g_2(n) \le (c_1 \cdot f_1(n)) \cdot (c_2 \cdot f_2(n)) = (c_1 \cdot c_2) \cdot (f_1(n) \cdot f_2(n))$. By the definition of O-notation, we conclude that $g_1(n) \cdot g_2(n) = O(f_1(n) \cdot f_2(n))$.

- **(ii). [5 marks]** What is the asymptotic (**Big-Oh**) complexity of the function $g(n) = (n^2 + \sqrt{n}) \cdot (n + \log n)$?

  Let $g_1(n) = n^2 + \sqrt{n}$, $g_2(n) = n + \log n$. Then $g_1(n) = O(n^2)$, $g_2(n) = O(n)$. According to the product rule, $g(n) = g_1(n) \cdot g_2(n) = O(n^2 \cdot n) = O(n^3)$.

- **(iii). [5 marks]** What is the asymptotic (**Big-Theta**) complexity of the function $g(n) = (n^3 + 3n^2 + 5) \cdot (n^2 + n^4)$?

  Let $g_1(n) = n^3+3n^2+5$, $g_2(n) = n^2+n^4$. Then $g_1(n) = \Theta(n^3)$, $g_2(n) = \Theta(n^4)$. According to the product rule, $g(n) = g_1(n) \cdot g_2(n) = \Theta(n^3 \cdot n^4) = \Theta(n^7)$.

- **(iv). [8 marks]** Let $f(n)$ and $g(n)$ be asymptotically nonnegative functions. Using the basic definition of Θ-notation, prove that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

  There exist n_1 and n_2 so that when $n \ge n_1$ and $n \ge n_2$, $f(n) \ge 0$ and $f(n) \ge 0$. We define $n_0 = \max(n_1, n_2)$. Then when $n \ge n_0$, we have

$$
\begin{aligned}
f(n) &\le \max(f(n), g(n)), \\
g(n) &\le \max(f(n), g(n)), \\
\frac{f(n) + g(n)}{2} &\le \max(f(n), g(n)), \\
\max(f(n), g(n)) &\le f(n) + g(n).
\end{aligned}
$$

  So we have $0 \le \frac{f(n)+g(n)}{2} \le \max(f(n), g(n)) \le f(n) + g(n)$. According to the definition of Θ-notation, there exist positive constants $c_1 = \frac{1}{2}$, $c_2 = 1$, and n_0 such that $0 \le c_1(f(n) + g(n)) \le \max(f(n), g(n)) \le c_2(f(n) + g(n))$. We conclude that $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.

- **Q3. [15 marks]** Answer the following questions about the linked list.
  - **(i). [5 marks]** Assume that we have a linked list: $6 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 9$ as shown in Fig. 1. Fill the values in the question marks.

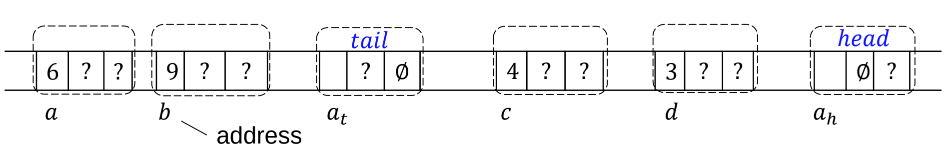

Fig. 1. Q3 (i)

Solution:

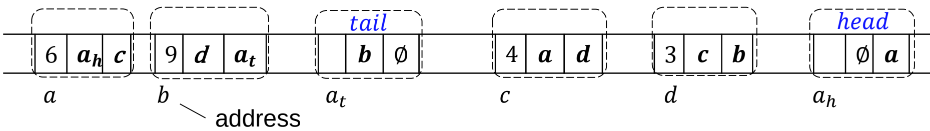

Fig. 2. Q3 (i) Solution

- **(ii). [5 marks]** We have a linked list: $2 \leftrightarrow 3 \leftrightarrow 5$ as shown in Fig. 3. Assume that we insert element 7, and the address of the node is a. Update the values after the insertion.

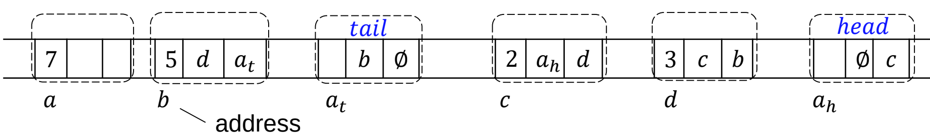

Fig. 3. Q3 (ii)

Solution:

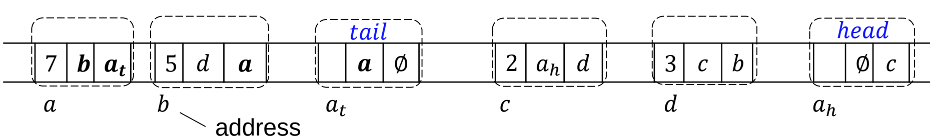

Fig. 4. Q3 (ii) Solution

- **(iii). [5 marks]** We have a linked list: $3 \leftrightarrow 6 \leftrightarrow 5 \leftrightarrow 8$ as shown in Fig. 5. Update the values after deleting the node storing element 8.

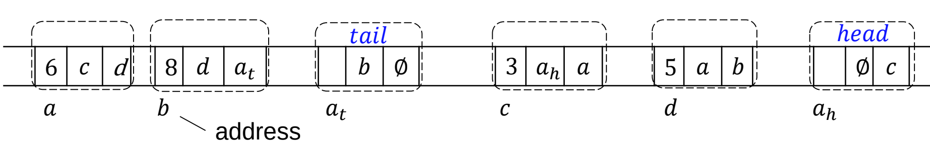

Fig. 5. Q3 (iii)

Solution:

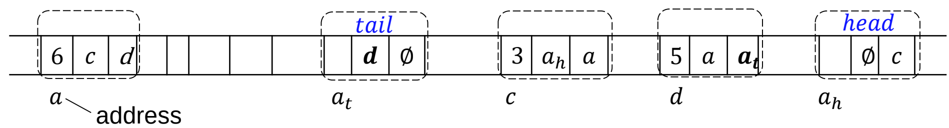

Fig. 6. Q3 (iii) Solution

- **Q4. [12 marks]** Use the master method to give asymptotic (**Big-Oh**) bounds for the following recurrences.
  - **(i). [4 marks]** $g(1) = c_0$, $g(n) \leq 4 \cdot g(n/4) + 1$.

    We have that $a = 4$, $b = 4$ and $\lambda = 0$. Since $log_b a > \lambda$, we have that: $g(n) = O(n^{\log_b a}) = O(n)$.
  - **(ii). [4 marks]** $g(1) = c_0$, $g(n) \leq 2 \cdot g(n/4) + 2\sqrt{n}$.

    We have that $a = 2$, $b = 4$ and $\lambda = \frac{1}{2}$. Since $log_b a = \lambda$, we have that: $g(n) = O(n^{\lambda} \log n) = O(\sqrt{n} \cdot \log n)$.
  - **(iii). [4 marks]** $g(1) = c_0$, $g(n) \leq 2 \cdot g(n/4) + 3n^2$.

    We have that $a = 2$, $b = 4$ and $\lambda = 2$. Since $log_b a < \lambda$, we have that: $g(n) = O(n^{\lambda}) = O(n^2)$.
- **Q5. [12 marks]** For the following functions, sort them in **nonincreasing** order of the growth rate. (Hint. If $f_1 = O(f_2)$, f_2 grows not slower than f_1.)

$$
\begin{aligned}
f_1(n) &= 2^{2^{100000}} \\
f_2(n) &= 2^{10000n} \\
f_3(n) &= \binom{n}{2} \\
f_4(n) &= n\sqrt{n}
\end{aligned}
$$

The correct order of these functions is $f_2(n) \geq f_3(n) \geq f_4(n) \geq f_1(n)$. The variable n never appears in the formula for $f_1(n)$, so despite the multiple exponentials, $f_1(n)$ is constant. Hence, it is asymptotically smaller than $f_4(n)$, which does grow with n. We may rewrite the formula for $f_4(n)$ to be $f_4(n) = n\sqrt{n} = n^{1.5}$. The value of $f_3(n) = \binom{n}{2}$ is given by the formula $n(n-1)/2$, which is $\Theta(n^2)$. Hence, $f_4(n) = n^{1.5} = O(n^2) = O(f_3(n))$. Finally, $f_2(n)$ is exponential, while $f_3(n)$ is quadratic, meaning that $f_3(n)$ is $O(f_2(n))$.

- **Q6. [16 marks]** Given that $T(1) = 1$, answer the following questions.
  - **(i). [8 marks]** Show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$.

We guess $T(n) \le c \cdot n^2$,

$$\begin{aligned} T(n) &\le c \cdot (n-1)^2 + n \\ &= c \cdot n^2 - 2c \cdot n + c + n \\ &= c \cdot n^2 + n \cdot (1 - 2c) + c \\ &\le c \cdot n^2, \end{aligned}$$

where the last step holds for $c > \frac{1}{2}$.

- **(ii). [8 marks]** Show that the solution of $T(n) = T(\lceil n/2 \rceil) + 1$ is $O(\log n)$.

  We guess $T(n) \le c \cdot \log(n-2)$,

$$\begin{aligned} T(n) &\le c \cdot \log(\lceil n/2 \rceil - 2) + 1 \\ &\le c \cdot \log(n/2 + 1 - 2) + 1 \\ &= c \cdot \log((n-2)/2) + 1 \\ &= c \cdot \log(n-2) - c \cdot \log 2 + 1 \\ &\le c \cdot \log(n-2), \end{aligned}$$

  where the last step holds for $c \ge 1$.

- **Q7. [9 marks]** Assume that $n = b^{y+1}$ and a, b, λ are constants such that $a \ge 1$, $b > 1$ and $\lambda \ge 0$. Given that $g(n) = c \cdot n^\lambda \left(1 + \frac{a}{b^\lambda} + \left(\frac{a}{b^\lambda}\right)^2 + \cdots + \left(\frac{a}{b^\lambda}\right)^y\right) + c_2 \cdot n^{\log_b a}$, discuss the asymptotic complexity of g(n) in terms of **Big-Omega** in the following cases. (Hint 1. For $x = 1$, $\sum_{i=0}^{n} x^i = n+1$. For $x \ne 1$, $\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}$. Hint 2. Definition of Big-Omega notation. Hint 3: $a^{\log_b n} = n^{\log_b a}$ and $b^{\log_b n} = n$.)

  - **(i). [3 marks]** Case 1: $\log_b a < \lambda$.

    Since $\log_b a < \lambda$, $a < b^\lambda$, $\frac{a}{b^\lambda} < 1$ and $y = \log_b n - 1$.

$$\begin{aligned} g(n) &\ge c \cdot n^\lambda \left(1 + \frac{a}{b^\lambda} + \left(\frac{a}{b^\lambda}\right)^2 + \cdots + \left(\frac{a}{b^\lambda}\right)^y\right) + c_2 \cdot n^{\log_b a} \\ &= c \frac{1 - \left(\frac{a}{b^\lambda}\right)^{y+1}}{1 - \frac{a}{b^\lambda}} \cdot n^\lambda + c_2 \cdot n^{\log_b a} \\ &\ge c \cdot n^\lambda + c_2 \cdot n^{\log_b a} \\ &= \Omega(n^\lambda) \end{aligned}$$

  - **(ii). [3 marks]** Case 2: $\log_b a = \lambda$.

Since $\log_b a = \lambda$, $a = b^\lambda$, $\frac{a}{b^\lambda} = 1$ and $y = \log_b n - 1$.

$$
\begin{aligned}
g(n) &\geq c \cdot n^\lambda \left(1 + \frac{a}{b^\lambda} + \left(\frac{a}{b^\lambda}\right)^2 + \cdots + \left(\frac{a}{b^\lambda}\right)^y\right) + c_2 \cdot n^{\log_b a} \\
&= c \cdot \log_b n \cdot n^\lambda + c_2 \cdot n^\lambda \\
&= \Omega(n^\lambda \cdot \log n)
\end{aligned}
$$

– **(iii). [3 marks]** Case 3: $\log_b a > \lambda$.

Since $\log_b a > \lambda$, $a > b^\lambda$, $\frac{a}{b^\lambda} > 1$ and $y = \log_b n - 1$.

$$
\begin{aligned}
g(n) &\geq c \cdot n^\lambda \left(1 + \frac{a}{b^\lambda} + \left(\frac{a}{b^\lambda}\right)^2 + \cdots + \left(\frac{a}{b^\lambda}\right)^y\right) + c_2 \cdot n^{\log_b a} \\
&= c \frac{1 - \left(\frac{a}{b^\lambda}\right)^{y+1}}{1 - \frac{a}{b^\lambda}} \cdot n^\lambda + c_2 \cdot n^{\log_b a} \\
&\geq c \cdot n^\lambda + c_2 \cdot n^{\log_b a} \\
&= \Omega(n^{\log_b a})
\end{aligned}
$$