Lab10-Turing Machine

Algorithm and Complexity (CS214), Xiaofeng Gao, Spring 2020.

∗ If there is any problem, please contact TA Yiming Liu.
∗ Name:Zehao Wang Student ID:518021910976 Email: davidwang200099@sjtu.edu.cn

1. Design a one-tape TM M that computes the function $f(x, y) = x \mod y$, where x and y are positive integers ($x > y$). The alphabet is $\{1, 0, \square, \triangleright, \triangleleft\}$, and the inputs are x 1's, $\square$ and y 1's. Below is the initial configuration for input $x = 7$ and $y = 3$. The result $z = f(x, y)$ should also be represented in the form of z 1's on the tape with the pattern of $\triangleright 111 \cdots 111 \triangleleft$.

Initial Configuration

	$\triangleright$	1	1	1	1	1	1	1	$\square$	1	1	1	$\triangleleft$	
	$\uparrow$													
	q_S													

(a) Please describe your design and then write the specifications of M in the form like $\langle q_S, \triangleright\rangle \to \langle q_1, \triangleright, R\rangle$. Explain the transition functions in detail.

(b) Please draw the state transition diagram.

(c) Show briefly and clearly the whole process from initial to final configurations for input $x = 7$ and $y = 3$. You may start like this:

$$(q_s, \underline{\triangleright}1111111\square 111\triangleleft) \vdash (q_1, \triangleright\underline{1}111111\square 111\triangleleft) \vdash^* (q_1, \triangleright 1111111\underline{\square}111\triangleleft) \vdash (q_2, \triangleright 1111111\square\underline{1}11\triangleleft)$$

(Note that for simplicity, we write $(q_1, \triangleright\underline{1}111111\square 111\triangleleft) \vdash^* (q_1, \triangleright 1111111\underline{\square}111\triangleleft)$ if the corresponding transaction repeats on multiple inputs with the same state.)

Solution.

(a) The basic idea is continuously trying to subtract y from x when $y > x$. When $y < x$, this new y is the result of $y \ mod \ x$.

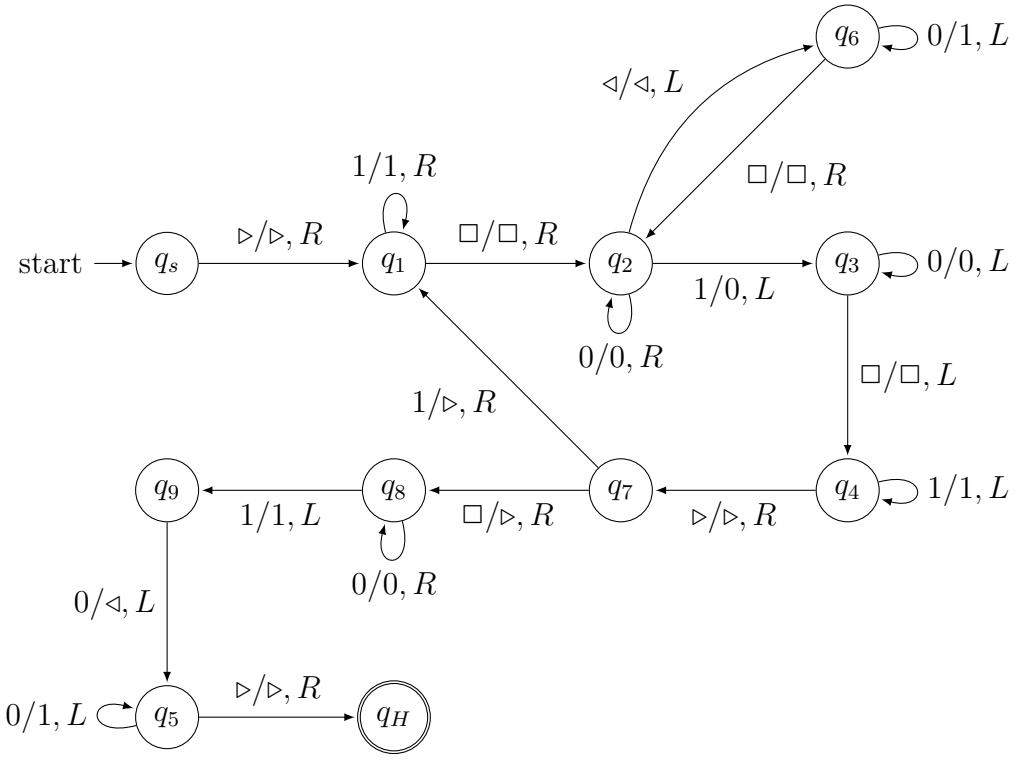
As for the operation on the Turing Machine, we first try to mark a 1 in y into 0, which denotes that we want to subtract 1 from x. Then we try to change a 1 in x into $\triangleright$, which means that we successfully subtract 1 from x. We repeat the operation above. When the 1's in y are all turned into 0, we need to resume y 0's into 1's. At this moment, we successfully subtract y from x.

Then we need to repeat the operation above until all the 1's in x are turned into $\triangleright$. This time, the number of 0's in y is $x \ mod \ y + 1$. So we need to turn them into 1 and use a pair of triangle to locate them.

The transition function is as follows.

$\langle q_S, \triangleright\rangle \to \langle q_1, \triangleright, R\rangle$ $\langle q_1, 1\rangle \to \langle q_1, 1, R\rangle$ $\langle q_2, \triangleleft\rangle \to \langle q_6, \triangleleft, L\rangle$
$\langle q_6, 0\rangle \to \langle q_6, 1, L\rangle$ $\langle q_6, \square\rangle \to \langle q_2, \square, R\rangle$ $\langle q_2, 1\rangle \to \langle q_3, 0, L\rangle$
$\langle q_2, 0\rangle \to \langle q_2, 0, R\rangle$ $\langle q_3, 0\rangle \to \langle q_3, 0, L\rangle$ $\langle q_3, \square\rangle \to \langle q_4, \square, L\rangle$
$\langle q_4, 1\rangle \to \langle q_4, 1, L\rangle$ $\langle q_4, \triangleright\rangle \to \langle q_7, \triangleright, R\rangle$ $\langle q_7, 1\rangle \to \langle q_1, \triangleright, R\rangle$
$\langle q_7, \square\rangle \to \langle q_8, \triangleright, R\rangle$ $\langle q_8, 0\rangle \to \langle q_8, 0, R\rangle$ $\langle q_8, 1\rangle \to \langle q_9, 1, L\rangle$
$\langle q_9, 0\rangle \to \langle q_5, \triangleleft, L\rangle$ $\langle q_5, 0\rangle \to \langle q_5, 1, R\rangle$ $\langle q_5, \triangleright\rangle \to \langle q_H, \triangleright, R$
$\langle q_1, \square\rangle \to \langle q_2, \square, R\rangle$

(b) This picture is drawn with TiKZ.



(c)

$(q_s, \underline{\triangleright}1111111\square111\triangleleft) \vdash (q_1, \triangleright\underline{1}111111\square111\triangleleft) \vdash^* (q_1, \triangleright1111111\underline{\square}111\triangleleft) \vdash (q_2, \triangleright1111111\square\underline{1}11\triangleleft)$

$\vdash (q_3, \triangleright1111111\underline{\square}011\triangleleft) \vdash (q_4, \triangleright111111\underline{1}\square011\triangleleft) \vdash^* (q_4, \underline{\triangleright}1111111\square011\triangleleft) \vdash (q_7, \triangleright\underline{1}111111\square011\triangleleft)$

$\vdash (q_1, \triangleright\triangleright\underline{1}11111\square011\triangleleft) \vdash^* (q_1, \triangleright\triangleright111111\underline{\square}011\triangleleft) \vdash (q_2, \triangleright\triangleright111111\square\underline{0}11\triangleleft) \vdash (q_2, \triangleright\triangleright111111\square0\underline{1}1\triangleleft)$

$\vdash (q_3, \triangleright\triangleright111111\square\underline{0}01\triangleleft) \vdash (q_3, \triangleright\triangleright111111\underline{\square}001\triangleleft) \vdash (q_4, \triangleright\triangleright11111\underline{1}\square001\triangleleft) \vdash^* (q_4, \triangleright\triangleright\underline{\triangleright}111111\square001\triangleleft)$

$\vdash (q_7, \triangleright\triangleright\underline{1}11111\square001\triangleleft) \vdash (q_1, \triangleright\triangleright\triangleright\underline{1}1111\square001\triangleleft) \vdash^* (q_1\triangleright\triangleright\triangleright11111\underline{\square}001\triangleleft) \vdash^* (q_2\triangleright\triangleright\triangleright11111\square\underline{0}01)$

$\vdash^* (q_2, \triangleright\triangleright\triangleright11111\square00\underline{1}\triangleleft) \vdash (q_3, \triangleright\triangleright\triangleright11111\square0\underline{0}0\triangleleft) \vdash^* (q_3, \triangleright\triangleright\triangleright11111\underline{\square}000\triangleleft) \vdash (q_4, \triangleright\triangleright\triangleright1111\underline{1}\square000\triangleleft)$

$\vdash^* (q_4, \triangleright\triangleright\triangleright\underline{\triangleright}1111\square000\triangleleft) \vdash (q_7, \triangleright\triangleright\triangleright\underline{1}1111\square000\triangleleft) \vdash (q_1, \triangleright\triangleright\triangleright\triangleright\underline{1}111\square000\triangleleft) \vdash^* (q_1, \triangleright\triangleright\triangleright\triangleright1111\underline{Box}000\triangleleft)$

$\vdash (q_2, \triangleright\triangleright\triangleright\triangleright1111\square\underline{0}00\triangleleft) \vdash^* (q_2, \triangleright\triangleright\triangleright\triangleright1111\square000\underline{\triangleleft}) \vdash (q_6, \triangleright\triangleright\triangleright\triangleright1111\square00\underline{0}\triangleleft) \vdash^* (q_6, \triangleright\triangleright\triangleright\triangleright1111\underline{\square}111\triangleleft)$

$$\vdash (q_2, \triangleright\triangleright\triangleright\triangleright 1111\square\underline{1}11\triangleleft) \vdash (q_3, \triangleright\triangleright\triangleright\triangleright 1111\underline{\square}011\triangleleft) \vdash (q_4, \triangleright\triangleright\triangleright\triangleright 111\underline{1}\square 011\triangleleft) \vdash^* (q_4, \triangleright\triangleright\triangleright\underline{\triangleright} 1111\square 011\triangleleft)$$

$$\vdash (q_7, \triangleright\triangleright\triangleright\triangleright\underline{1}1111\square 011\triangleleft) \vdash (q_1, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{1}11\square 011\triangleleft) \vdash^* (q_1, \triangleright\triangleright\triangleright\triangleright\triangleright 111\underline{\square}011\triangleleft) \vdash (q_2, \triangleright\triangleright\triangleright\triangleright\triangleright 111\square\underline{0}11\triangleleft)$$

$$\vdash (q_2, \triangleright\triangleright\triangleright\triangleright\triangleright 111\square 0\underline{1}1\triangleleft) \vdash (q_3, \triangleright\triangleright\triangleright\triangleright\triangleright 111\square\underline{0}01\triangleleft) \vdash (q_3, \triangleright\triangleright\triangleright\triangleright\triangleright 111\underline{\square}001\triangleleft) \vdash (q_4, \triangleright\triangleright\triangleright\triangleright\triangleright 11\underline{1}\square 001\triangleleft)$$

$$\vdash^* (q_4, \triangleright\triangleright\triangleright\triangleright\underline{\triangleright} 111\square 001\triangleleft) \vdash (q_7, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{1}11\square 001\triangleright) \vdash (q_1, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright\underline{1}1\square 001\triangleleft) \vdash^* (q_1, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 11\underline{\square}001\triangleleft)$$

$$\vdash (q_2, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 11\square\underline{0}01\triangleleft) \vdash^* (q_2, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 11\square 00\underline{1}\triangleleft) \vdash (q_3, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 11\square 0\underline{0}0\triangleleft) \vdash^* (q_3, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 11\underline{\square}000\triangleleft)$$

$$\vdash (q_4, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 1\underline{1}\square 000\triangleleft) \vdash^* (q_4, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{\triangleright} 11\square 000\triangleleft) \vdash (q_7, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright\underline{1}1\square 000\triangleleft)$$

$$\vdash (q_1, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright\underline{1}\square 000\triangleleft) \vdash (q_1, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 1\underline{\square}000\triangleleft) \vdash (q_2, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 1\square\underline{0}00\triangleleft) \vdash^* (q_2, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 1\square 000\underline{\triangleleft})$$

$$\vdash (q_6, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 1\square 00\underline{0}\triangleleft) \vdash^* (q_6, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 1\underline{\square}111\triangleleft) \vdash (q_2, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 1\square\underline{1}11\triangleleft) \vdash (q_3, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright 1\underline{\square}011\triangleleft)$$

$$\vdash (q_4, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright\underline{1}\square 011\triangleleft) \vdash (q_4, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{\triangleright} 1\square 011\triangleleft) \vdash (q_7, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{1}\square 011\triangleleft) \vdash (q_1, \triangleright\triangleright\triangleright\triangleright\triangleright\triangleright\underline{\square}011\triangleleft)$$

$$\vdash (q_2, \triangleright\triangleright\triangleright\triangleright\triangleright\square\underline{0}11\triangleleft) \vdash (q_2, \triangleright\triangleright\triangleright\triangleright\triangleright\square 0\underline{1}1\triangleleft) \vdash (q_3, \triangleright\triangleright\triangleright\triangleright\triangleright\square\underline{0}01\triangleleft) \vdash (q_3, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{\square}001\triangleleft)$$

$$\vdash (q_4, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{\triangleright}\square 001\triangleleft) \vdash (q_7, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{\square}001\triangleleft) \vdash (q_8, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{0}01\triangleleft) \vdash^* (q_8, \triangleright\triangleright\triangleright\triangleright\triangleright 00\underline{1}\triangleleft)$$

$$\vdash (q_9, \triangleright\triangleright\triangleright\triangleright\triangleright 00\underline{1}\triangleleft) \vdash (q_9, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{0}\triangleleft 0\triangleleft) \vdash (q_5, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{\triangleright} 1\triangleleft 0\triangleleft) \vdash (q_H, \triangleright\triangleright\triangleright\triangleright\triangleright\underline{1}\triangleleft 0\triangleleft)$$

2. Assume there's a Turing Machine M using alphabet $\Gamma : \{\triangleright, \square, a, b, \cdots, z\}$. We can simulate M by a Turing Machine $\tilde{M}$ using alphabet $\tilde{\Gamma} : \{\triangleright, \square, 0, 1\}$. Please transform the instruction $\langle q, i\rangle \to \langle q', j, R\rangle$ in M into its corresponding form in $\tilde{M}$.

Solution. There are 26 letters in English. Therefore to denote them in bits, each letter requires 5 bits. Assume that $a = 00000$ and $z = 11001$ and the letters are denoted with big endian method, that is, the most significant bit is on the left and the least significant one is on the right.

Then to find out the current letter is i, the head should check these five bits. This require extra interim states. Then the head should change i into j. This may also require extra interim states. i is 01000 and j is 01001. In this problem, we are luckily enough to have to change only one bit and thus fewer inerim states are necessary. However, on some unlucky occasions, the head may have to read the 5 bits, change the 5 bits and then move on to the next letter. These kind of cases require more interim states.

In this problem, we need to introduce only 4 interim states. The head should check the 5 bits one by one to figure out the current letter is i and then change the least significant bit from 0 to 1.

The transition function can be listed as follows.

$\langle q, 0\rangle \to \langle q_1, 0, R\rangle \quad \langle q_1, 1\rangle \to \langle q_2, 1, R\rangle \quad \langle q_2, 0\rangle \to \langle q_3, 0, R\rangle$

$\langle q_3, 0\rangle \to \langle q_4, 0, R\rangle \quad \langle q_4, 0\rangle \to \langle q', 1, R\rangle$ □

3. **Wireless Data Broadcast System.** In a Wireless Data Broadcast System (WDBS), data items are repeatedly broadcasted in cycle on different channels. Denote $D = \{d_1, d_2, \cdots, d_k\}$ as data items, each d_i with length l_i (as time units), and $\mathbf{C} = \{C_1, C_2, \cdots, C_n\}$ as broadcasting channels. Fig. 1 illustrates a WDBS with 25 data items and 4 channels. Once a channel finishes broadcasting current cycle, it will repeat these data again as a new cycle. E.g., a possible broadcasting sequence of C_1 could be $\{d_6, d_{12}, d_1, d_{18}, d_7, d_6, d_{12}, d_1, d_{18}, d_7, \cdots\}$

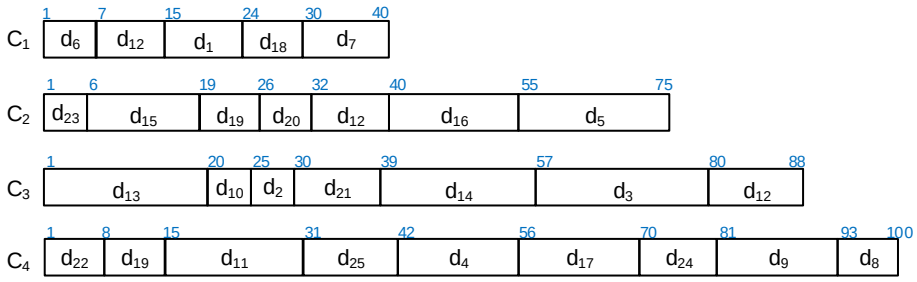


Figure 1: An Example Scenario of Wireless Data Broadcast System.

If a mobile client requires a subset of data items $D_q \subseteq D$ from this WDBS, he/she must access onto one channel, wait for the appearance of one required item, and switch to another channel if necessary. Each "switch" requires one time slot. For example, Lucien wants to download $\{d_1, d_3, d_5\}$, as shown in Fig. 2. He firstly accesses onto C_1 at time slot 1, then download d_1, d_3 respectively during time slots 2 to 5, and then switch to C_3 at time slot 6 (note that he cannot download d_5 from C_2 because of the switch constraint), and download d_5 during time slots 7 to 8. We define *access latency* as the period when a client starts downloading, till the time he/she finishes. As a result, the overall access latency for Lucien is 7 in this example.

Each operation (download/wait/switch) needs energy consumption. To conserve energy, a client hopes to use minimum amount of energy to download all required items in D_q, which means that he/she waits to minimize both access latency and switch numbers. Unfortunately, these two objectives conflict with each other naturally. Fig. 3 exhibits such a scenario. To download $D_q = \{d_1, d_2, d_3, d_4\}$, if we start from C_2, in Option 1 we can switch to C_1 for d_1 immediately after downloading d_3, return back to C_2 for d_4, and to C_1 again for d_2. Such

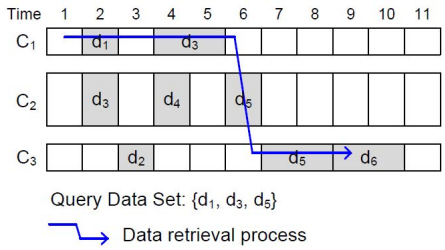


Figure 2: An Example Scenario of Query of a Client.

option costs 3 switches and 7 access latency. While in Option 2, we stay at C_2 lazily for d_3 and d_4, and then switch to C_1 for d_2 and d_1. Such option costs 1 switches and 12 access latency.

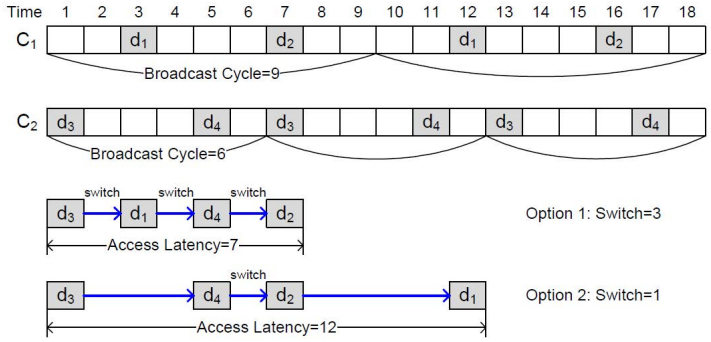


Figure 3: Confliction between Access Latency and Switch Number.

Once we want to minimize two conflictive objectives simultaneously, we have three possible ways (similar as Segmented Least Squares told in Dynamic Programming Lecture). Now it is your turn to complete the formulation of this optimization, we name it as Minimum Constraint Data Retrieval Problem (MCDR), with the following sub-questions.

(a) If we add an additional switch parameter h, please define the MCDR (Version 1) completely as a search problem.

(b) If we add an additional latency parameter t, please define the MCDR (Version 2) completely as a search problem.

(c) If we set dimensional parameters α to switch number, and β to access latency, we can combine two objectives together linearly as a new concept "cost". Please define the Minimum Cost Data Retrieval Problem (MCDR, Version 3) correspondingly.

(d) Please give the decision versions of sub-questions (a), (b) and (c).

Solution. (a) Given a data set $D = \{d_1, d_2, \cdots, d_n\}$ and a channel set $C = \{c_1, c_2, \cdots, c_m\}$. Each piece of data d_i has a length L_i and lies in a channel c_i. We want to retrieve a subset: D_q, of D from these channels. Try to find a method which has minimal data access latency and requires at most h switches between the channels.

(b) Given a data set $D = \{d_1, d_2, \cdots, d_n\}$ and a channel set $C = \{c_1, c_2, \cdots, c_m\}$. Each piece of data d_i has a length L_i and lies in a channel c_i. We want to retrieve a subset: D_q, of D from these channels. Try to find a method which requires minimal number of switches and has a data access latency of at most t.

(c) Given a data set $D = \{d_1, d_2, \cdots, d_n\}$ and a channel set $C = \{c_1, c_2, \cdots, c_m\}$. Each piece of data d_i has a length L_i and lies in a channel c_i. We want to retrieve a subset:

D_q, of D from these channels. Define $cost = \alpha \times switch\ number + \beta \times access\ latency$. Try to find a method which has minimal cost.

(d)

- Question (a):Given a data set $D = \{d_1, d_2, \cdots, d_n\}$ and a channel set $C = \{c_1, c_2, \cdots, c_m\}$. Each piece of data d_i has a length L_i and lies in a channel c_i. We want to retrieve a subset: D_q, of D from these channels. Is there a method which has a data access latency not more than k and requires at most h switches between the channels?
- Question (b):Given a data set $D = \{d_1, d_2, \cdots, d_n\}$ and a channel set $C = \{c_1, c_2, \cdots, c_m\}$. Each piece of data d_i has a length L_i and lies in a channel c_i. We want to retrieve a subset: D_q, of D from these channels. Is there a method which requires not more than k switches and has a data access latency of at most t?
- Question (c):Given a data set $D = \{d_1, d_2, \cdots, d_n\}$ and a channel set $C = \{c_1, c_2, \cdots, c_m\}$. Each piece of data d_i has a length L_i and lies in a channel c_i. We want to retrieve a subset: D_q, of D from these channels. Define $cost = \alpha \times switch\ number + \beta \times access\ latency$. Is there a method whose cost is not more than c_0?

□