Q1

Consider a demand-paged virtual memory system with M page frames. Assume initially all frames are empty. Given a page reference string with length L and N distinctive page numbers (N < L),

a) What are the lower and upper bounds of page faults that may be generated by such a reference string if the FIFO page replacement algorithm is used?

b) Will you have different answers if the LRU (Least-Recently-Used), instead of FIFO, page replacement algorithm is used?

Q1 (a)

What are the lower and upper bounds of page faults that may be generated by such a reference string if the FIFO page replacement algorithm is used?

Hints: Consider two cases: 1) If N <= M, 2) If N > M.

Answer:

If N <= M, the lower and upper bounds are both N.

→ Once N distinct pages are loaded into the memory, no more page faults may occur.

If N>M, the lower bound is N and upper bound is L.

→ Can you construct the reference strings?

Q1 (a) -- **If N>M**

The lower bound is N

→ The reference string:

P1 P2 … PM P1 P2 … PM … … P1 P2 … PM P_{M+1} P_{M+2} … PN

Example (N=3, L=7, M=2): P1 P2 P1 P2 P1 P2 P3

The upper bound is L.

→ The reference string:

P1 P2 … PN P1 P2 … PN … … P1 P2 … PN

Example (N=3, L=7, M=2): P1 P2 P3 P1 P2 P3 P1

Q1 (b)

Will you have different answers if the LRU (Least-Recently-Used), instead of FIFO, page replacement algorithm is used?

→ No. The same as FIFO.

Q2

Suppose that a process is allocated four page frames in a demand-paged virtual memory system. The time of loading and the time of last access are as shown in the table below.

Frame #	Page #	Time Loaded	Time Last Accessed
0	1	60	161
1	0	130	160
2	3	26	162
3	2	20	163

Given the above memory state before the fault and the remaining page reference string of the process: 4, 0, 0, 0, 2, 4, 2, 1, 0, 3, 2, answer the following two questions.

a) Calculate how many page faults will be generated if Least-Recently-Used (LRU) algorithm is used. Show your workings.

b) Assuming that the size of the working set window is four, calculate the number of page faults that would occur if only the pages in the working set were loaded into the memory instead of using a fixed allocation of four frames. Show your workings.

Q2 (a)

Ref String		4	0	2	4	2	1	0	3	2
	2	4	0	2	4	2	1	0	3	2
	3	2	4	0	2	4	2	1	0	3
	1	3	2	4	0	0	4	2	1	0
LRU Page	0	1	3	3	3	3	0	4	2	1
Out Page		0	1				3		4	

Number of page faults: 4

Q2 (b)

Frame #	Page #	Time Loaded	Time Last Accessed
0	1	60	161
1	0	130	160
2	3	26	162
3	2	20	163

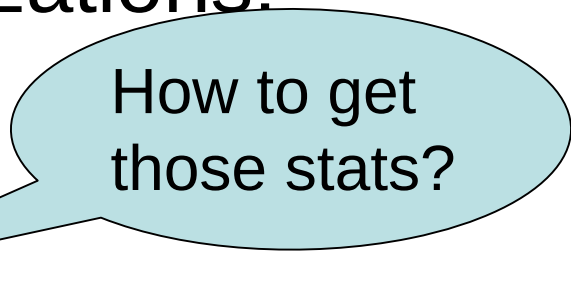
Ref String	Working Set				Page Fault
	0	1	3	2	
4	1	3	2	4	P
0	3	2	4	0	P
0	2	4	0		
0	4	0			
2	0	2			P
4	0	2	4		P
2	0	2	4		
1	2	4	1		P
0	4	2	1	0	P
3	2	1	0	3	P
2	1	0	3	2	

Number of page faults: 7

Q3

3. Consider a demand-paging system with the following time-measured utilizations:

CPU utilization 20%
Paging disk 97.7%
Other I/O devices 5%



Which (if any) of the following will (probably) improve CPU utilization?

Q3: improve CPU utilization?

CPU utilization 20%
Paging disk 97.7%
Other I/O devices 5%

a. Install a faster CPU. No.
b. Install a bigger paging disk. No.
c. Increase the degree of multiprogramming. No.
d. Decrease the degree of multiprogramming. Yes.
e. Install more main memory. Yes.
f. Install a faster hard disk or multiple hard disks. Probably yes.
g. Increase the page size. Depends on memory access pattern.

Q3

(f) A possible improvement, for as the disk bottleneck is removed by faster response and more throughput to the disks, the CPU will get more data more quickly.

(g)
- Increasing the page size will result in fewer page faults if data is being accessed sequentially
- If data access is more or less random, more paging actions could occur because fewer pages can be kept in memory and more data is transferred per page fault.