

Push_swap 평가표

평가 항목

Memory leaks

Error management

이 섹션에서는 push_swap의 오류 관리를 평가합니다.

하나 이상의 실패가 발생할 경우 이에 대한 점수가 부여되지 않습니다.

부분적인 다음 칸으로 이동하세요.

- Run push_swap 매개변수가 숫자가 아니라면 "Error"가 화면에 표시 되어야 합니다.
- Run push_swap 숫자가 중복이 된다면 "Error"가 화면에 표시 되어야 합니다.
- Run push_swap 숫자가 int범위를 넘어간다면 "Error".
- Run push_swap 매개변수가 없다면 화면에 아무거나 표시해서 빨리 돌려주세요.

Push_swap - Identity test

이 섹션에서는 이미 정렬된 목록이 주어졌을 때 push_swap의 동작을 평가합니다.

다음 세 가지 테스트를 실행합니다.

하나 이상 불합격할 경우 이 섹션에 대한 점수가 부여되지 않습니다. 다음 칸으로 이동하세요.

- ./push_swap 42". 프로그램에 아무 것도 표시하지 않아야 합니다(0 지침).
- ./push_swap 0 1 2 3". 프로그램에 아무 것도 표시되지 않아야 합니다(0 지침).
- ./push_swap 0 1 2 3 4 5 6 7 8 9". 프로그램에 아무 것도 표시되지 않아야 합니다(0 지침).

Push_swap - Simple version

- ARG="2 1 0"; ./push_swap \$ARG | ./checker_OS \$ARG".

검사 프로그램이 "OK"로 표시되는지 확인합니다.

push_swap의 명령 목록 크기가 2 OR 3인 경우. 그렇지 않으면 테스트가 실패합니다.

Another simple version

- ARG="1 5 2 4 3"; ./push_swap \$ARG | ./checker_OS \$ARG".

검사 프로그램이 "OK"로 표시되는지 확인합니다. and that the

push_contract의 명령 목록 크기가 12를 초과하지 않습니다. 지시사항 목록의 크기가 8이면 Kudos.

- ARG="<5 random values>"; ./push_swap \$ARG | ./checker_OS \$ARG" 및 자리 표시자를 5개의 랜덤 유효 값으로 바꿉니다.

검사 프로그램이 "OK"를 표시하고 push_swap의 명령 목록 크기가 12를 초과하지 않는지 확인합니다. 그렇지 않으면 이 테스트가 실패합니다. 이 척도에 포함된 테스트에서만 올바르게 응답하도록 프로그램이 개발되지 않았는지 구체적으로 확인해야 합니다. 검증하기 전에 이 테스트를 여러 순열로 몇 번 반복해야 합니다.

Push_swap - Middle version

- Run "\$>ARG="<100 random values>"; ./push_swap \$ARG | ./checker_OS \$ARG" and replace the placeholder by 100 random 자리 표시자를 100개의 랜덤 유효 값으로 바꿉니다. 검사 프로그램이 "OK"을 표시하고 지시사항 목록의 크기를 확인합니다. 에 따라 점수를 부여한다.

- less than 700: 5
- less than 900: 4
- less than 1100: 3
- less than 1300: 2
- less than 1500: 1

이 척도에 포함된 테스트에서만 올바르게 응답하도록 프로그램이 개발되지 않았는지 구체적으로 확인해야 합니다. 검증하기 전에 이 테스트를 여러 순열로 몇 번 반복해야 합니다.

Push_swap - Advanced version

- Run "\$>ARG="<500 random values>"; ./push_swap \$ARG | ./checker_OS \$ARG" and replace the placeholder by 500 random valid values (One is not called John/Jane Script for nothing). Check that the checker program displays "OK" and that the size of the list of instructions

- less than 5500: 5
- less than 7000: 4
- less than 8500: 3
- less than 10000: 2
- less than 11500: 1

You'll have to specifically check that the program wasn't developed to only answer correctly on the test included in this scale.

You should repeat this test couple of times with several permutations before you validate it.

Bonus

Checker program - Error management (push_swap 에러 처리와 동일)

- Run checker with non numeric parameters. The program must display "Error".

- Run checker with a duplicate numeric parameter. The program must display "Error".

- Run checker with only numeric parameters including one greater than MAXINT. The program must display "Error".

- Run checker without any parameters. The program must not display anything and give the prompt back.

- Run checker with valid parameters, and write an action that doesn't exist during the instruction phase. The program must display "Error".

- 유효한 매개 변수를 사용하여 검사기를 실행하고 명령 단계에서 작업 전/후에 하나 또는 여러 공백을 사용하여 작업을 작성합니다. The program must display "Error".

Checker program - False tests

Don't forget to press CTRL+D to stop reading during the instruction phase.

- Run checker with the following command "\$>./checker 0 9 1 8 2 7 3 6 4 5" then write the following valid action list "[sa, pb, rrr]". Checker should display "KO".

Checker program - Right tests

- Run checker with the following command "\$>./checker 0 1 2" then press CTRL+D without writing any instruction. The program should display "OK".

- Run checker with the following command "\$>./checker 0 9 1 8 2" then write the following valid action list "[pb, ra, pb, ra, sa, ra, pa, pa]". The program should display "OK".