

ESCALA PARA O PROJETO PUSH_SWA (/PROJECTS/42CURSUS-PUSH_SWAP)

Você deve avaliar 1 aluno desta equipe.



Repositório Git

git@vogsphere.42porto.com:vogsphere/intra-uuid-52a5d6dd-5c8d-4c

Introdução

Por favor, siga as seguintes regras:

- Mantenha-se educado, cortês, respeitoso e construtivo durante todo o processo de avaliação. O bem-estar da comunidade depende disso.
- Identifique possíveis disfunções no projeto do aluno ou grupo avaliado. Dedique um tempo para discutir e debater os problemas que possam ter sido identificados.
- Considere que pode haver diferenças na forma como seus colegas interpretaram as instruções do projeto e o escopo de suas funcionalidades. Mantenha sempre a mente aberta e avalie os trabalhos com a maior honestidade possível. A pedagogia só é eficaz se a avaliação por pares for levada a sério.

Diretrizes

- Avalie apenas o trabalho entregue no repositório Git do aluno ou grupo avaliado.
- Certifique-se de que o repositório Git pertence ao(s) aluno(s) e que o projeto é o esperado. Além disso, certifique-se de que o comando 'git clone' seja executado em uma pasta vazia.
- Verifique cuidadosamente se nenhum alias malicioso foi usado para enganá-lo e fazer com que avalie algo que não corresponde ao conteúdo do repositório oficial.
- Para evitar surpresas e, se aplicável, revise em conjunto quaisquer scripts usados para facilitar a avaliação (scripts para testes ou automação).
- Se você não concluiu a tarefa que irá avaliar, leia todo o conteúdo antes de iniciar o processo de avaliação.
- Use as opções disponíveis para reportar um repositório vazio, um programa que não funciona, um erro de norma, plágio, etc. Nesses casos, o processo de avaliação termina e a nota final é 0, ou -42 em caso de plágio. No entanto, exceto em casos de plágio, os alunos são fortemente encorajados a revisar em conjunto o trabalho entregue, a fim de identificar quaisquer erros que não devem ser repetidos no futuro.
- Você também deve verificar a ausência de vazamentos de memória. Qualquer memória alocada no heap deve ser liberada corretamente antes do término da execução. Você pode usar qualquer uma das ferramentas disponíveis no computador, como leaks, valgrind ou e_fence. Se forem detectados vazamentos de memória, selecione a flag apropriada.

Anexos

assunto.pdf (<https://cdn.intra.42.fr/pdf/pdf/189089/en.subject.pdf>)

verificador_Mac (https://cdn.intra.42.fr/document/document/42927/checker_Mac)

[verificador_linux \(https://cdn.intra.42.fr/document/document/42928/checker_linux\)](https://cdn.intra.42.fr/document/document/42928/checker_linux)

Parte obrigatória

Lembrete: Durante a defesa, não são permitidas falhas de segmentação, travamentos inesperados prematuros ou saídas não controladas. Caso contrário, a nota final será 0. Utilize a flag apropriada aplica durante toda a defesa.

README.md Verificação de conformidade

O repositório contém um arquivo README.md na raiz e inclui todos os seguintes itens?

- A primeira linha está em itálico e formatada exatamente como: *Este projeto foi criado como 42 por <login1>*.
- Uma seção de "Descrição" explicando o propósito do projeto e fornecendo uma breve visão
- Uma seção de "Instruções" com detalhes relevantes sobre compilação, instalação e/ou execução
- Uma seção "Recursos" listando referências (documentação, tutoriais, etc.) e explicando como especificar para quais tarefas e em quais partes do projeto.

Sim

Não

Vazamentos de memória

Durante a defesa, monitore o uso de memória do push_swap (por exemplo, usando o comando top) para detectar anomalias e garantir que toda a memória alocada seja liberada corretamente. Se houver um ou mais vazamentos de memória, a nota final será 0.

Sim

Não

Gestão de erros

Nesta seção, avaliaremos o gerenciamento de erros do push_swap. Se pelo menos um erro falhar, nenhum ponto será atribuído a esta seção. Passe para a próxima.

- Execute o push_swap com parâmetros não numéricos. O programa deve exibir "Erro" seguido de um '\n' na mensagem de erro padrão.
- Execute a função push_swap com um parâmetro numérico duplicado. O programa deve exibir "Erro" seguido de um '\n' na mensagem de erro padrão.
- Execute push_swap com apenas parâmetros numéricos, incluindo um maior que MAXINT. O programa deve exibir "Erro" seguido de um '\n' na mensagem de erro padrão.
- Execute o push_swap sem parâmetros. O programa não deve exibir nada e deve retornar o prompt.

Sim

Não

Push_swap - Teste de identidade

Nesta seção, avaliaremos o comportamento de `push_swap` quando uma lista já estiver ordenada. Execute os seguintes testes. Se pelo menos um falhar, nenhum ponto será atribuído a esta seção. Passe para a próxima.

- Execute o seguinte comando "\$>/push_swap 42". O programa não deve exibir nada (0 instruções).
- Execute o seguinte comando "\$>/push_swap 2 3". O programa não deve exibir nada (0 instruções).
- Execute o seguinte comando "\$>/push_swap 0 1 2 3". O programa não deve exibir nada (0 instruções).
- Execute o seguinte comando "\$>/push_swap 0 1 2 3 4 5 6 7 8 9". O programa não deve exibir nada (0 instruções).
- Execute o seguinte comando "\$>/push_swap <Entre 0 e 9 valores ordenados escolhidos aleatoriamente>". O programa não deve exibir nada (instrução 0).

Sim Não

Push_swap - Versão simples

Se os testes a seguir falharem, nenhum ponto será atribuído a esta seção. Passe para a próxima. Use o arquivo binário de verificação fornecido nos anexos.

- Execute "\$>ARG="2 1 0"; ./push_swap \$ARG | ./checker_OS \$ARG". Verifique se o programa de verificação exibe "OK" e se o tamanho da lista de instruções de push_swap é 2 OU 3. Caso contrário, o teste falha.
- Execute "\$>ARG=<Entre 0 e 3 valores escolhidos aleatoriamente>"; ./push_swap \$ARG | ./checker_OS \$ARG". Verifique se o programa de verificação exibe "OK" e se o tamanho da lista de instruções de push_swap está entre 0 e 3. Caso contrário, o teste falha.

 Sim Não

Outra versão simples

Execute os dois testes a seguir. Se pelo menos um deles falhar, nenhum ponto será atribuído a esta seção. Passe para o próximo. Use o arquivo binário de verificação fornecido nos anexos.

- Execute "\$>ARG="1 5 2 4 3"; ./push_swap \$ARG | ./checker_OS \$ARG". Verifique se o programa de verificação exibe "OK" e se o tamanho da lista de instruções de push_swap não excede 12. Parabéns se a lista de instruções contiver apenas 8 etapas.
- Execute "\$>ARG=<5 valores aleatórios>"; ./push_swap \$ARG | ./checker_OS \$ARG" e substitua o marcador por 5 valores válidos aleatórios. Verifique se o programa de verificação exibe "OK" e se o tamanho da lista de instruções de `push_swap` não é maior que 12. Caso contrário, este teste falha. Você precisará verificar especificamente se o programa não foi desenvolvido para responder corretamente apenas ao teste incluído nesta escala. Você deve repetir este teste algumas vezes com várias permutações antes de validá-lo.

 Sim Não

Push_swap - Versão intermediária

Se o teste a seguir falhar, nenhum ponto será atribuído a esta seção. Passe para a próxima. Use o arquivo binário de verificação fornecido nos anexos.

- Execute "\$>ARG=<100 valores aleatórios>"; ./push_swap \$ARG | ./checker_OS \$ARG" e o marcador de posição por 100 valores aleatórios válidos. Verifique se o programa de verificação se o tamanho da lista de instruções está correto. Atribua os pontos de acordo com a pontuação:
 - menos de 700: 5
 - menos de 900: 4
 - menos de 1100: 3
 - menos de 1300: 2
 - Menos de 1500: 1 Você precisará verificar especificamente se o programa não foi responder corretamente apenas ao teste incluído nesta escala. Você deve repetir este vezes com diferentes permutações antes de validá-lo.

Avalie de 0 (reprovado) a 5 (excelente).



Push_swap - Versão avançada

Se o teste a seguir falhar, nenhum ponto será atribuído a esta seção. Passe para a próxima. Use o arquivo binário de verificação fornecido nos anexos.

Intra Projects push_swap Edit

- Execute "\$>ARG=<500 valores aleatórios>; ./push_swap \$ARG | ./checker_OS \$ARG" e s marcador por 500 valores aleatórios válidos (O script não se chama John/Jane Script à toa). programa de verificação exibe "OK" e se o número de instruções está dentro dos seguintes l
 - menos de 5500: 5
 - menos de 7000: 4
 - menos de 8500: 3
 - menos de 10000: 2
- Menos de 11500: 1 Você terá que verificar especificamente se o programa não foi des responder corretamente apenas ao teste incluído nesta escala. Você deve repetir este vezes com várias permutações antes de validá-lo.

Avalie de 0 (reprovado) a 5 (excelente).

**Bônus**

Lembrete: Durante a defesa, não serão tolerados erros de segmentação (segfault) nem qualquer o encerramento inesperado, prematuro, descontrolado ou imprevisível do programa. Caso contrário, 0. Utilize a flag apropriada. Esta regra permanece ativa durante toda a defesa. Avaliaremos seus b se a parte obrigatória for EXCELENTE. Isso significa que você deve concluir a parte obrigatória do seu gerenciamento de erros precisa ser impecável, mesmo em casos de uso incorreto ou inadequ obrigatoria não receber a nota máxima durante esta defesa, a seção de bônus será completamente

Programa de verificação - Gerenciamento de erros

Nesta seção, avaliaremos o gerenciamento de erros do verificador. Se pelo menos um erro for detectado, nenhum ponto será atribuído a esta seção. Passe para a próxima.

- Execute o verificador com parâmetros não numéricos. O programa deve exibir "Erro" seguido de uma '\n' na mensagem de erro padrão.
- Execute o verificador com um parâmetro numérico duplicado. O programa deve exibir "Erro" seguido de uma '\n' na mensagem de erro padrão.
- Execute o verificador com apenas parâmetros numéricos, incluindo um maior que MAXINT. O programa deve exibir "Erro" seguido de um '\n' na mensagem de erro padrão.
- Execute o verificador sem parâmetros. O programa não deve exibir nada e deve retornar o prompt.
- Execute o verificador com parâmetros válidos e escreva uma ação que não exista durante a fase de instruções. O programa deve exibir "Erro" seguido de um '\n' no erro padrão.
- Execute o verificador com parâmetros válidos e escreva uma ação com um ou mais espaços antes e/ou depois da ação durante a fase de instruções. O programa deve exibir "Erro" seguido de um '\n' no erro padrão.

 Sim

 Não
Programa de verificação - Testes falsos

Nesta seção, avaliaremos a capacidade do verificador de gerenciar uma lista de instruções que não a ordena. Execute os dois testes a seguir. Se pelo menos um falhar, nenhum ponto será atribuído a esta seção. Passe para o próximo teste.

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]". The checker should display "KO".
- Run checker with a valid list as parameter of your choice then write a valid instruction list that doesn't order the integers. The checker should display "KO". You'll have to specifically check that the program wasn't developed to only

Intra Projects push_swap Edit

answer correctly on the test included in this scale. You should repeat this test couple of times with several permutations before you validate it.

 Yes

 No
Checker program - Right tests

In this section, we'll evaluate the checker's ability to manage a list of instructions that sort the list. Execute the following 2 tests. If at least one fails, no points will be awarded for this section. Move to the next one.

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

- 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".
- Run checker with a valid list as parameter of your choice then write a valid instruction list that correctly orders the integers. Checker must display "OK". 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.

 Yes

 No
Ratings

Don't forget to check the flag corresponding to the defense

 Ok

 Outstanding project

 Empty work

 Incomplete work

 Invalid compilation

 Norme

 Cheat

 Concerning situation

 Leaks

 Forbidden function

 Can't support /
Conclusion

Leave a comment on this evaluation (2048 chars max)

Finish evaluation