

Programação Competitiva

Aula Extra - Stress Test

Emanuel Juliano Moraes Silva

Universidade Federal de Minas Gerais

10 de novembro de 2020



Stress Test - O que é

Stress Test - O que é

- Stress Test é uma ferramenta de debug que compara diversos casos aleatórios e verifica se seu programa está gerando a saída esperada.

Stress Test - O que é

- Stress Test é uma ferramenta de debug que compara diversos casos aleatórios e verifica se seu programa está gerando a saída esperada.
- O stress test é usado quando se precisa de casos teste para debugar o código, ou para conferir se ele está certo sem receber WA.

Stress Test - O que é

- Stress Test é uma ferramenta de debug que compara diversos casos aleatórios e verifica se seu programa está gerando a saída esperada.
- O stress test é usado quando se precisa de casos teste para debugar o código, ou para conferir se ele está certo sem receber WA.
- Como ele precisa de um gerador de casos aleatórios e de um código que sempre retorna a saída certa, o stress test pode ser inviável, mas em alguns casos é muito útil:

Stress Test - O que é

- Stress Test é uma ferramenta de debug que compara diversos casos aleatórios e verifica se seu programa está gerando a saída esperada.
- O stress test é usado quando se precisa de casos teste para debugar o código, ou para conferir se ele está certo sem receber WA.
- Como ele precisa de um gerador de casos aleatórios e de um código que sempre retorna a saída certa, o stress test pode ser inviável, mas em alguns casos é muito útil:
 - ▶ Problemas de guloso com ordenação

Stress Test - O que é

- Stress Test é uma ferramenta de debug que compara diversos casos aleatórios e verifica se seu programa está gerando a saída esperada.
- O stress test é usado quando se precisa de casos teste para debugar o código, ou para conferir se ele está certo sem receber WA.
- Como ele precisa de um gerador de casos aleatórios e de um código que sempre retorna a saída certa, o stress test pode ser inviável, mas em alguns casos é muito útil:
 - ▶ Problemas de guloso com ordenação
 - ▶ Problemas de programação dinâmica

Stress Test - O que é

- Stress Test é uma ferramenta de debug que compara diversos casos aleatórios e verifica se seu programa está gerando a saída esperada.
- O stress test é usado quando se precisa de casos teste para debugar o código, ou para conferir se ele está certo sem receber WA.
- Como ele precisa de um gerador de casos aleatórios e de um código que sempre retorna a saída certa, o stress test pode ser inviável, mas em alguns casos é muito útil:
 - ▶ Problemas de guloso com ordenação
 - ▶ Problemas de programação dinâmica
 - ▶ Problemas de simulação

Stress Test - O que é

- Stress Test é uma ferramenta de debug que compara diversos casos aleatórios e verifica se seu programa está gerando a saída esperada.
- O stress test é usado quando se precisa de casos teste para debugar o código, ou para conferir se ele está certo sem receber WA.
- Como ele precisa de um gerador de casos aleatórios e de um código que sempre retorna a saída certa, o stress test pode ser inviável, mas em alguns casos é muito útil:
 - ▶ Problemas de guloso com ordenação
 - ▶ Problemas de programação dinâmica
 - ▶ Problemas de simulação
 - ▶ Problemas que envolvem contas com índice

Stress Test - O que é

- Stress Test é uma ferramenta de debug que compara diversos casos aleatórios e verifica se seu programa está gerando a saída esperada.
- O stress test é usado quando se precisa de casos teste para debugar o código, ou para conferir se ele está certo sem receber WA.
- Como ele precisa de um gerador de casos aleatórios e de um código que sempre retorna a saída certa, o stress test pode ser inviável, mas em alguns casos é muito útil:
 - ▶ Problemas de guloso com ordenação
 - ▶ Problemas de programação dinâmica
 - ▶ Problemas de simulação
 - ▶ Problemas que envolvem contas com índice
 - ▶ Problema de grafo

Stress Test - O que precisamos

Stress Test - O que precisamos

- `a.cpp`: um código na complexidade certa, mas que possui bugs.

Stress Test - O que precisamos

- `a.cpp`: um código na complexidade certa, mas que possui bugs.
- `a2.cpp`: um código com complexidade errada, mas que retorna a resposta certa.

Stress Test - O que precisamos

- `a.cpp`: um código na complexidade certa, mas que possui bugs.
- `a2.cpp`: um código com complexidade errada, mas que retorna a resposta certa.
- `gen.cpp`: um gerador de casos aleatórios.

Stress Test - O que precisamos

- `a.cpp`: um código na complexidade certa, mas que possui bugs.
- `a2.cpp`: um código com complexidade errada, mas que retorna a resposta certa.
- `gen.cpp`: um gerador de casos aleatórios.
- `stress.sh`: um script bash que automatiza os testes.

Stress Test - O que precisamos

- `a.cpp`: um código na complexidade certa, mas que possui bugs.
- `a2.cpp`: um código com complexidade errada, mas que retorna a resposta certa.
- `gen.cpp`: um gerador de casos aleatórios.
- `stress.sh`: um script bash que automatiza os testes.
- Obs: como `stress.sh` é um arquivo em bash, ele precisa ser executado de forma especial, existem dois modos de fazer isso:

Stress Test - O que precisamos

- `a.cpp`: um código na complexidade certa, mas que possui bugs.
- `a2.cpp`: um código com complexidade errada, mas que retorna a resposta certa.
- `gen.cpp`: um gerador de casos aleatórios.
- `stress.sh`: um script bash que automatiza os testes.
- Obs: como `stress.sh` é um arquivo em bash, ele precisa ser executado de forma especial, existem dois modos de fazer isso:
 - ▶ `bash stress.sh` (executando de uma vez)

Stress Test - O que precisamos

- **a.cpp**: um código na complexidade certa, mas que possui bugs.
- **a2.cpp**: um código com complexidade errada, mas que retorna a resposta certa.
- **gen.cpp**: um gerador de casos aleatórios.
- **stress.sh**: um script bash que automatiza os testes.
- Obs: como **stress.sh** é um arquivo em bash, ele precisa ser executado de forma especial, existem dois modos de fazer isso:
 - ▶ `bash stress.sh` (executando de uma vez)
 - ▶ `chmod +x stress.sh` (gerando um arquivo executável `./stress.sh`)

Stress Test - stress.sh

Stress Test - stress.sh

```
1  make a a2 gen || exit 1
2  for ((i = 1; ; i++)) do
3      ./gen $i > in
4      ./a < in > out
5      ./a2 < in > out2
6      if (! cmp -s out out2) then
7          echo "--> entrada:"
8          cat in
9          echo "--> saida1:"
10         cat out
11         echo "--> saida2:"
12         cat out2
13         break;
14     fi
15     echo $i
16 done
```

Stress Test - gen.cpp

Stress Test - gen.cpp

```
1  int rand(int a, int b){ // valor aleatório no intervalo [a, b]
2      return rand()%(b-a+1) + a;
3  }
4
5  int main(int argc, char** argv){
6      srand(atoi(argv[1])); // define a seed para o rand()
7
8      int n = rand(1, 10), m = rand(1, 10);
9      cout << n << " " << m << endl;
10 }
```

Stress Test - gen.cpp

```
1  int rand(int a, int b){ // valor aleatório no intervalo [a, b]
2      return rand()%(b-a+1) + a;
3  }
4
5  int main(int argc, char** argv){
6      srand(atoi(argv[1])); // define a seed para o rand()
7
8      int n = rand(1, 10), m = rand(1, 10);
9      cout << n << " " << m << endl;
10 }
```

● Entrada

- ▶ ./gen 1
- ▶ ./gen 2

● Saída

- ▶ 4 7
- ▶ 1 10

Exercício resolvido: No Time for Dragons - gen.cpp

```
1  int rand(int a, int b){
2      return rand()%(b-a+1) + a;
3  }
4  int main(int argc, char**argv){
5      srand(atoi(argv[1]));
6
7      int n = rand(1, 5);
8      cout << n << endl;
9      for(int i=0; i <n; i++){
10         int a = rand(1, 10), b=rand(1, 10);
11         if(a<b) swap(a, b);
12         cout << a << " " << b << endl;
13     }
14     cout << endl;
15     exit(0);
16 }
```

Exercício resolvido: No Time for Dragons - a2.cpp

```
1  int n, sum = 0; cin >> n;
2  vector<ii> v(n);
3  for(auto &i: v) cin >> i.f >> i.s, sum += i.f;
4
5  for(int i=1; i<=sum; i++){
6      while(true){ // testa todas as permutações com next_permutation
7          int at = i;
8          bool show = true;
9          for(auto [a, b] : v){
10              if(at<a) show = false;
11              at -= b;
12          }
13          if(show) return cout << i << endl, 0;
14          if(!next_permutation(v.begin(), v.end())) break;
15      }
16 }
```

Material e Links

- [stress.sh](#)
- [Vídeo do Errichto](#)
- [Stress Test para questão de grafos](#)
- [Lista de Exercícios](#)