**Thí nghiệm**

**Case A: Kiểm tra độ chính xác của kỹ thuật Group Testing.**

**Case B: So sánh độ hiệu quả của kỹ thuật PSS-SFL so với các kỹ thuật Spectrum-based đã biết.**

Phần 1. Đo độ chính xác (hiệu quả) của 5 kỹ thuật:

1. Group Testing (chỉ xét việc xác định đúng block lỗi hay không)
2. Tarantula
3. Ochiai
4. Jaccard
5. FSS-SFL

Chương trình 1: Tính giá trị tuyệt đối

Chương trình đúng:

|  |
| --- |
| int abs(int n) {  if (n >= 0) {  return n;  } else {  return -n;  }  } |

+1.Chương trình sai:

|  |
| --- |
| int abs(int n) {  if (n >= 0) {  return n;  } else {  return n;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kỹ thuật |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

+2.Chương trình sai:

|  |
| --- |
| int abs(int n) {  if (n >= 0) {  return -n;  } else {  return -n;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

+3.Chương trình sai:

|  |
| --- |
| int abs(int n) {  if (n >= 0) {  return n + 1;  } else {  return -n;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

+4.Chương trình sai:

|  |
| --- |
| int abs(int n) {  if (n >= 0) {  n = n + 1;  return 1;  } else {  return -n;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

+5.Chương trình sai:

|  |
| --- |
| int abs(int n) {  if (n >= 5) {  return n;  } else {  return -n;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

+6.Chương trình sai:

|  |
| --- |
| int abs(int n) {  if (n < 0) {  return n;  } else {  return -n;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

+7.Chương trình sai:

|  |
| --- |
| int abs(int n) {  if (n == 0) {  return n;  } else {  return -n;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

Chương trình 2: Kiểm tra tính chẵn lẻ

Chương trình đúng:

|  |
| --- |
| int even(int n) {  if (n % 2 == 0) {  return 1;  } else {  return 0;  }  } |

+1.Chương trình sai:

|  |
| --- |
| int even(int n) {  if (n % 2 == 0) {  return 0;  } else {  return 0;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

+2.Chương trình sai:

|  |
| --- |
| int even(int n) {  if (n % 2 == 0) {  return 1;  } else {  return 1;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

+3.Chương trình sai:

|  |
| --- |
| int even(int n) {  if (n % 2 != 0) {  return 1;  } else {  return 0;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

Chương trình 3: Trả về số lớn hơn

Chương trình đúng:

|  |
| --- |
| int max(int a, int b) {  if (a >= b) {  return a;  } else {  return b;  }  } |

+1.Chương trình sai:

|  |
| --- |
| int max(int a, int b) {  int ret = 0;  if (a >= b) {  ret = b; // CAU LENH LOI 3  } else {  ret = b;  }  return ret;  } |

Kết quả:

* Group Testing:
* Fault Localization

+2.Chương trình sai:

|  |
| --- |
| int max(int a, int b) {  int ret = 0;  if (a >= b) {  ret = a;  } else {  ret = a; // CAU LENH LOI 4  a = a + 1;  }  return ret;  } |

Kết quả:

* Group Testing:
* Fault Localization

+3.Chương trình sai:

|  |
| --- |
| int max(int a, int b) {  if (a < b) {  return a;  } else {  return b;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

+4.Chương trình sai:

|  |
| --- |
| int max(int a, int b) {  if (a >= b + 5) {  return a;  } else {  return b;  }  } |

Kết quả:

* Group Testing:
* Fault Localization

Chương trình 4: Tính giai thừa

Chương trình đúng:

|  |
| --- |
| int fact(int n) {  int r = 1;  int i = 1;  while (i <= n) {  r = r \* i;  i = i + 1;  }  return r;  } |

+1.Chương trình sai:

|  |
| --- |
| int fact(int n) {  int r = 0; // CAU LENH LOI 1  int i = 1;  while (i <= n && r != 0) { // CAU LENH NAY DUNG  r = r \* i;  i = i + 1;  }  return r;  } |

Kết quả:

* Group Testing:
* Fault Localization

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kỹ thuật |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

+2.Chương trình sai:

|  |
| --- |
| int fact(int n) {  int r = 1;  int i = 1;  while (i <= n) {  r = r + i; // CAU LENH LOI 4  i = i + 1;  }  return r;  } |

Kết quả:

* Group Testing:
* Fault Localization

+3.Chương trình sai:

|  |
| --- |
| int fact(int n) {  int r = 1;  int i = 1;  while (i <= n) {  r = r \* i;  i = i + 1;  }  return r + 1; // CAU LENH LOI 6  } |

Kết quả:

* Group Testing:
* Fault Localization

+4.Chương trình sai:

|  |
| --- |
| int fact(int n) {  int r = 1;  int i = 1;  while (i <= n + 2) {  r = r \* i;  i = i + 1;  }  return r + 1; // CAU LENH LOI 6  } |

Kết quả:

* Group Testing:
* Fault Localization

+5.Chương trình sai:

|  |
| --- |
| int fact(int n) {  int r = 0; // CAU LENH SAI 1  while (n > 1) {  r = r \* n;  n = n - 1;  }  return r;  } |

Kết quả:

* Group Testing:
* Fault Localization

+6.Chương trình sai:

|  |
| --- |
| int fact(int n) {  int r = 1;  while (n > 1) {  r = r + n; // CAU LENH LOI 3  n = n - 1;  }  return r;  } |

Kết quả:

* Group Testing:
* Fault Localization

+7.Chương trình sai:

|  |
| --- |
| int fact(int n) {  int r = 1;  while (n > 3) { // CAU LENH LOI 2  r = r \* n;  n = n - 1;  }  return r;  } |

Kết quả:

* Group Testing:
* Fault Localization

Chương trình 5: Selection sort

Chương trình đúng:

|  |
| --- |
| int sort(int n, int a[])  {  int i = 0;  while (i < n - 1) {  int min = i;  int j = i + 1;  while (j < n) {  if (a[j] < a[min]) {  min = j;  }  j = j + 1;  }  int temp = a[i];  a[i] = a[min];  a[min] = temp;  i = i + 1;  }  return a;  } |

+1.Chương trình sai:

|  |
| --- |
| int sort(int n, int a[])  {  int i = 0;  while (i < n - 1) {  int min = i;  int j = i + 1;  while (j < n) {  // if (a[j] < a[min])  if (a[j] > a[min]) { // CAU LENH LOI 6  min = j;  }  j = j + 1;  }  int temp = a[i];  a[i] = a[min];  a[min] = temp;  i = i + 1;  }  return a;  } |

Kết quả:

* Group Testing:
* Fault Localization

+2.Chương trình sai:

|  |
| --- |
| int sort(int n, int a[])  {  int i = 0;  while (i < n - 1) {  int min = i;  int j = i + 1;  while (j < n) {  if (a[j] < a[min]) {  min = j;  }  j = j + 1;  }  int temp = a[i];  a[i] = a[min];  a[min] = temp + 1; // CAU LENH LOI 11  i = i + 1;  }  return a;  } |

Kết quả:

* Group Testing:
* Fault Localization

Chương trình 6: Bubble sort

Chương trình đúng:

|  |
| --- |
| int sort(int n, int a[])  {  int swapped = 1;  int i = n - 1;  while(i > 0 && swapped == 1) {  int j = 0;  swapped = 0;  while(j < i) {  if(a[j] > a[j+1]) {  int temp = a[j];  a[j] = a[j+1];  a[j+1] = temp;  swapped = 1;  }  j = j + 1;  }  i = i - 1;  }  return a;  } |

+1.Chương trình sai:

|  |
| --- |
| int sort(int n, int a[])  {  int swapped = 1;  int i = n - 1;  while(i > 0 && swapped == 1) {  int j = 0;  swapped = 0;  while(j < i) {  if(a[j] > a[j+1]) {  int temp = a[j];  a[j] = a[j+1];  a[j+1] = temp + 1;  swapped = 1;  }  j = j + 1;  }  i = i - 1;  }  return a;  } |

Kết quả:

* Group Testing:
* Fault Localization

+2.Chương trình sai:

|  |
| --- |
| int sort(int n, int a[])  {  int swapped = 1;  int i = n - 1;  while(i > 0 && swapped == 1) {  int j = 0;  swapped = 0;  while(j < i) {  if(a[j] > a[j+1] \* 2) {  int temp = a[j];  a[j] = a[j+1];  a[j+1] = temp;  swapped = 1;  }  j = j + 1;  }  i = i - 1;  }  return a;  } |

Kết quả:

* Group Testing:
* Fault Localization

Chương trình 7: Assignment 1 môn Kỹ thuật lập trình, học kỳ 2 năm học 2009-2010, Khoa KH&KTMT, Đại học Bách Khoa TPHCM

Chương trình đúng:

|  |
| --- |
| float main(int HP1, int HP2, int Q1, int Q2, int d)  {  float fOut = 0.0;  float P1 = HP1;  float P2 = HP2;    int h = (HP1 + HP2) % 100;  if((d==1000) && ( h!=99 )&&( HP1!=999 ))  fOut= 0.00;  else {  if((HP1==777) && (Q1 < Q2) && (HP2!=888)) //Aramis tham chien  d = 201;    if ((Q1 > 2\*Q2) && (d>=800)) {  P1 = HP1 + (Q1 - Q2) \* d \* 1.0 / (Q1 + Q2);  }  else if(2\*Q1 < Q2 && d <= 200) {  P2 = HP2 + (Q2 - Q1)\*(1000 - d) \* 1.0 / (Q1 + Q2); // phep chia so thuc  }    if ((HP1 == 888) && (HP2 != 888)) //Porthos tham chien  h = 10 \* h;    fOut=(P1+h-P2+1000)/2000.0; // phep chia so thuc    if (HP2==888) //de Jussac tham chien  {  if(HP1==999) //d'Artagnan tham chien  fOut = 1.00;  else if((HP1!=777) && (HP1!=888) && (HP1!=900))  fOut = 0.01;  }    if ((HP1==900)&&(HP2!=888)) //Athos tham chien  if((Q1>Q2)||(fOut<0.5))  fOut=0.50;  }  return fOut;  } |

+1.Chương trình sai:

|  |
| --- |
| float main(int HP1, int HP2, int Q1, int Q2, int d)  {  float fOut = 0.0;  float P1 = HP1;  float P2 = HP2;    int h = (HP1 + HP2) % 100;  if((d==1000) && ( h!=99 )&&( HP1!=999 ))  fOut= 0.00;  else {  if((HP1==777) && (Q1 < Q2) && (HP2!=888)) //Aramis tham chien  d = 201;    if ((Q1 > 2\*Q2) && (d>=800)) {  P1 = HP1 + (Q1 - Q2) \* d \* 1.0/ (Q1 + Q2);  }  else if(2\*Q1 < Q2 && d <= 200) {  P2 = HP2 + (Q2 - Q1)\*(1000 - d)\*1.0 / (Q1 + Q2);  }    if ((HP1 == 888) && (HP2 != 888)) //Porthos tham chien  h = 10 \* h;    fOut=(P1+h-P2+1000)/2000.0;  // CAU LENH LOI 16  if (HP2!=888) //de Jussac tham chien  //if (HP2==888)  {  if(HP1==999) //d'Artagnan tham chien  fOut = 1.00;  else if((HP1!=777) && (HP1!=888) && (HP1!=900))  fOut = 0.01;  }    if ((HP1==900)&&(HP2!=888)) //Athos tham chien  if((Q1>Q2)||(fOut<0.5))  fOut=0.50;  }  return fOut;  } |

Kết quả:

* Group Testing:
* Fault Localization

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kỹ thuật |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

+2.Chương trình sai:

|  |
| --- |
| float main(int HP1, int HP2, int Q1, int Q2, int d)  {  float fOut = 0.0;  float P1 = HP1;  float P2 = HP2;    int h = (HP1 + HP2) % 100;  if((d==1000) && ( h!=99 )&&( HP1!=999 ))  fOut= 0.00;  else {  // CAU LENH LOI 7  if((HP1==777) || (Q1 < Q2) && (HP2!=888)) //Aramis tham chien  //if((HP1==777) && (Q1 < Q2) && (HP2!=888)) //Aramis tham chien  d = 201;    if ((Q1 > 2\*Q2) && (d>=800)) {  P1 = HP1 + (Q1 - Q2) \* d \* 1.0 / (Q1 + Q2);  }  else if(2\*Q1 < Q2 && d <= 200) {  P2 = HP2 + (Q2 - Q1)\*(1000 - d) \* 1.0 / (Q1 + Q2);  }    if ((HP1 == 888) && (HP2 != 888)) //Porthos tham chien  h = 10 \* h;    fOut=(P1+h-P2+1000)/2000.0;    if (HP2==888) //de Jussac tham chien  {  if(HP1==999) //d'Artagnan tham chien  fOut = 1.00;  else if((HP1!=777) && (HP1!=888) && (HP1!=900))  fOut = 0.01;  }    if ((HP1==900)&&(HP2!=888)) //Athos tham chien  if((Q1>Q2)||(fOut<0.5))  fOut=0.50;  }  return fOut;  } |

Kết quả:

* Group Testing:
* Fault Localization

+3.Chương trình sai:

|  |
| --- |
| float main(int HP1, int HP2, int Q1, int Q2, int d)  {  float fOut = 0.0;  float P1 = HP1;  float P2 = HP2;    int h = (HP1 + HP2) % 100;  if((d==1000) && ( h!=99 )&&( HP1!=999 ))  fOut= 0.00;  else {  if((HP1==777) && (Q1 < Q2) && (HP2!=888)) //Aramis tham chien  d = 201;    if ((Q1 > 2\*Q2) && (d>=800)) {  P1 = HP1 + (Q1 - Q2) \* d \* 1.0 / (Q1 + Q2);  }  else if(2\*Q1 < Q2 && d <= 200) {  P2 = HP2 + (Q2 - Q1)\*(1000 - d) \* 1.0 / (Q1 + Q2); // phep chia so thuc  }  // CAU LENH LOI 13  if ((HP1 == 888) || (HP2 != 888)) //Porthos tham chien  //if ((HP1 == 888) && (HP2 != 888)) //Porthos tham chien  h = 10 \* h;    fOut=(P1+h-P2+1000)/2000.0; // phep chia so thuc    if (HP2==888) //de Jussac tham chien  {  if(HP1==999) //d'Artagnan tham chien  fOut = 1.00;  else if((HP1!=777) && (HP1!=888) && (HP1!=900))  fOut = 0.01;  }    if ((HP1==900)&&(HP2!=888)) //Athos tham chien  if((Q1>Q2)||(fOut<0.5))  fOut=0.50;  }  return fOut;  } |

Kết quả:

* Group Testing:
* Fault Localization

+4.Chương trình sai:

|  |
| --- |
| float main(int HP1, int HP2, int Q1, int Q2, int d)  {  float fOut = 0.0;  float P1 = HP1;  float P2 = HP2;    int h = (HP1 + HP2) % 100;  if((d==1000) || ( h!=99 )||( HP1!=999 )) // CAU LENH LOI 5  fOut= 0.00;  else {  if((HP1==777) && (Q1 < Q2) && (HP2!=888)) //Aramis tham chien  d = 201;    if ((Q1 > 2\*Q2) && (d>=800)) {  P1 = HP1 + (Q1 - Q2) \* d \* 1.0 / (Q1 + Q2);  }  else if(2\*Q1 < Q2 && d <= 200) {  P2 = HP2 + (Q2 - Q1)\*(1000 - d) \* 1.0 / (Q1 + Q2); // phep chia so thuc  }    if ((HP1 == 888) && (HP2 != 888)) //Porthos tham chien  h = 10 \* h;    fOut=(P1+h-P2+1000)/2000.0; // phep chia so thuc    if (HP2==888) //de Jussac tham chien  {  if(HP1==999) //d'Artagnan tham chien  fOut = 1.00;  else if((HP1!=777) && (HP1!=888) && (HP1!=900))  fOut = 0.01;  }    if ((HP1==900)&&(HP2!=888)) //Athos tham chien  if((Q1>Q2)||(fOut<0.5))  fOut=0.50;  }  return fOut;  } |

Kết quả:

* Group Testing:
* Fault Localization

+5.Chương trình sai:

|  |
| --- |
| float main(int HP1, int HP2, int Q1, int Q2, int d)  {  float fOut = 0.0;  float P1 = HP1;  float P2 = HP2;    int h = (HP1 + HP2) % 100;  if((d==1000) && ( h!=99 )&&( HP1!=999 ))  fOut= 0.00;  else {  if((HP1==777) && (Q1 < Q2) && (HP2!=888)) //Aramis tham chien  d = 201;    if ((Q1 > 2\*Q2) && (d>=800)) {  P1 = HP1 + (Q1 - Q2) \* d / (Q1 + Q2); // CAU LENH LOI 10  }  else if(2\*Q1 < Q2 && d <= 200) {  P2 = HP2 + (Q2 - Q1)\*(1000 - d) \* 1.0 / (Q1 + Q2); // phep chia so thuc  }    if ((HP1 == 888) && (HP2 != 888)) //Porthos tham chien  h = 10 \* h;    fOut=(P1+h-P2+1000)/2000.0; // phep chia so thuc    if (HP2==888) //de Jussac tham chien  {  if(HP1==999) //d'Artagnan tham chien  fOut = 1.00;  else if((HP1!=777) && (HP1!=888) && (HP1!=900))  fOut = 0.01;  }    if ((HP1==900)&&(HP2!=888)) //Athos tham chien  if((Q1>Q2)||(fOut<0.5))  fOut=0.50;  }  return fOut;  } |

Kết quả:

* Group Testing:
* Fault Localization