تکلیف سوم

بخش اول

تمرین ۴ بخش ۸.۱

a) clauses are a، b and c.

b)

- $P_a = ((True \vee b) \iff c) \oplus ((False \vee b) \iff c)$

  $= (True \iff c) \oplus (b \iff c)$

  $= c \oplus (b \iff c)$

  $= (c \wedge \neg(b \iff c)) \vee (\neg c \wedge (b \iff c))$

  $= (c \wedge \neg b \wedge c) \vee (c \wedge b \wedge \neg c) \vee (\neg c \wedge b \wedge c) \vee (\neg c \wedge \neg b \wedge \neg c)$

  $= (c \wedge \neg b) \vee (\neg c \wedge \neg b)$

  $= \neg b \wedge (c \vee \neg c)$

  $= \neg b \wedge True$

  $= \neg b$

- $P_b = \neg a$ (similar to P_a)

- $P_c = ((a \vee b) \iff True) \oplus ((a \vee b) \iff False)$

  $= (a \vee b) \oplus \neg(a \vee b)$

  $= True$

c)

	a	b	c	$(a \vee b) \iff c$	P_a	P_b	P_c
۱	T	T	T	T	F	F	T
۲	T	T	F	F	F	F	T
۳	T	F	T	T	T	F	T
۴	T	F	F	F	T	F	T
۵	F	T	T	T	F	T	T
۶	F	T	F	F	F	T	T
۷	F	F	T	F	T	T	T
۸	F	F	F	T	T	T	T

d)

- GACC pairs for a: $(3,7),(3,8),(4,7),(4,8)$
- GACC pairs for b: $(5,7),(5,8),(6,7),(6,8)$
- GACC pairs for c: $(1,2),(1,4),(1,6),(1,8),(3,2),(3,4),(3,6),(3,8),$
  $(5,2),(5,4),(5,6),(5,8),(8,2),(7,4),(7,6),(7,8)$

e)

- CACC pairs for a: $(3,7),(4,8)$
- CACC pairs for b: $(5,7),(6,8)$
- CACC pairs for c: $(1,2),(1,4),(1,6),(3,2),(3,4),(3,6),(5,2),(5,4),(5,6),(7,8)$

f)

- RACC pairs for a: $(3,7),(4,8)$
- RACC pairs for b: $(5,7),(6,8)$
- RACC pairs for c: $(1,2),(3,4),(5,6),(7,8)$

بخش دوم

$a = i < size$

$b = j < i$

$c = intervals[i].begin < intervals[j].end$

$d = intervals[i].end > intervals[j].begin$

$e = rows[j] >= rows[i]$

$P = a \wedge b \wedge c \wedge d \wedge e$

پرسش اول

طبق جدول زیر برای CACC pair ها داریم:

$a = (1, 17), b = (1, 9), c = (1, 5), d = (1, 3), e = (1, 2)$

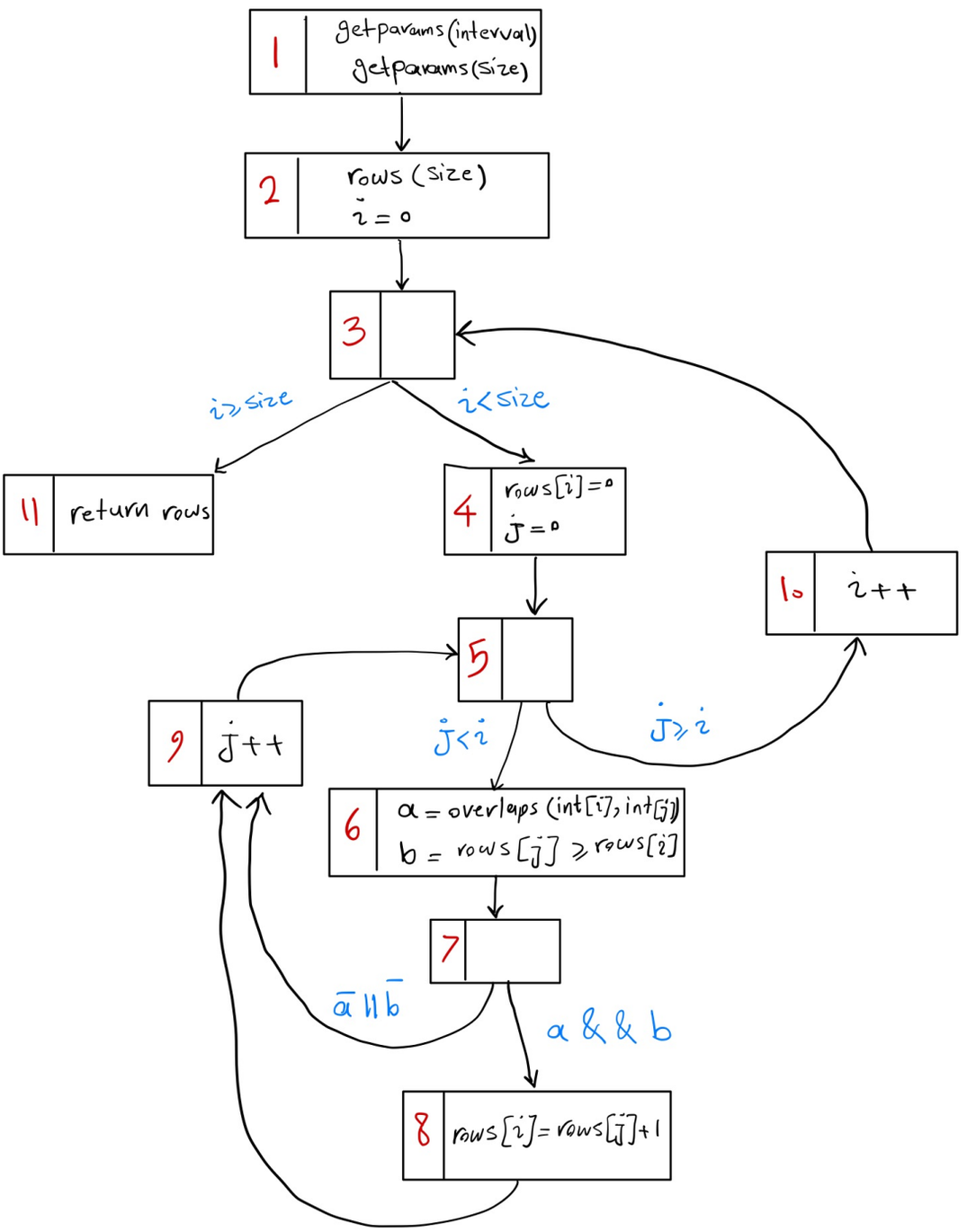
ولی سطر های 9 و 17 infeasible هستند و بنابراین تست $intervals = \{\{6, 7\}, \{10, 12\}, \{7, 9\}, \{6, 12\}\}, size = 4$ همه ی سطرهای $1, 2, 3, 5$ را پوشش میدهد.

	a	b	c	d	e	$(a \wedge b \wedge c \wedge d \wedge e$	P_a	P_b	P_c	P_d	P_e
۱	T	T	T	T	T	T	T	T	T	T	T
۲	T	T	T	T	F						T
۳	T	T	T	F	T					T	
۴	T	T	T	F	F						
۵	T	T	F	T	T				T		
۶	T	T	F	T	F						
۷	T	T	F	F	T						
۸	T	T	F	F	F						
۹	T	F	T	T	T			T			
۱۰	T	F	T	T	F						
۱۱	T	F	T	F	T						
۱۲	T	F	T	F	F						
۱۳	T	F	F	T	T						
۱۴	T	F	F	T	F						
۱۵	T	F	F	F	T						
۱۶	T	F	F	F	F						
۱۷	F	T	T	T	T		T				
۱۸	F	T	T	T	F						
۱۹	F	T	T	F	T						
۲۰	F	T	T	F	F						
۲۱	F	T	F	T	T						
۲۲	F	T	F	T	F						
۲۳	F	T	F	F	T						
۲۴	F	T	F	F	F						
۲۵	F	F	T	T	T						
۲۶	F	F	T	T	F						
۲۷	F	F	T	F	T						
۲۸	F	F	T	F	F						
۲۹	F	F	F	T	T						
۳۰	F	F	F	T	F						
۳۱	F	F	F	F	T						
۳۲	F	F	F	F	F						

پرسش دوم

با توجه به گراف زیر، مسیر طی شده به این صورت است:

$[1, 2, 3, 4, 5, 10, 3, 4, 5, 6, 7, 9, 5, 10, 3, 4, 5, 6, 7, 9, 5, 6, 7, 9, 5, 10, 3, 4, 5, 6, 7, 8, 9, 5, 6, 7, 9, 5, 6, 7, 9, 5, 10, 3, 11]$



پرسش سوم

مسیرهایی که قرمز شده اند infeasible هستند. مسیرهای سبزرنگ پوشش داده میشوند و سایر مسیرها پوشش داده نمیشوند.

[1, 2, 3, 4, 5, 10, 3, 4, 5, 6, 7, 9, 5, 10, 3, 4, 5, 6, 7, 9, 5, 6, 7, 9, 5, 10, 3, 4, 5, 6, 7, 8, 9, 5, 6, 7, 9, 5, 6, 7, 9, 5, 10, 3, 11]

Variable	DU-pair	DU-path
intervals	(1, 6)	[1, 2, 3, 4, 5, 6]
size	(1, 2)	[1, 2]
	(1, (3, 11))	[1, 2, 3, 11]
	(1, (3, 4))	[1, 2, 3, 4]
rows	(2, 6)	no def-clear path
	(2, 8)	no def-clear path
	(2, 11)	[2, 3, 11]
	(4, 6)	[4, 5, 6]
	(4, 8)	[4, 5, 6, 7, 8]
	(4, 11)	[4, 5, 10, 3, 11]
	(8, 6)	[8, 9, 5, 6]
	(8, 8)	[8, 9, 5, 6, 7, 8]
	(8, 11)	[8, 9, 5, 10, 3, 11]
i	(2, 4)	[2, 3, 4]
	(2, 6)	[2, 3, 4, 5, 6]
	(2, 8)	[2, 3, 4, 5, 6, 7, 8]
	(2, 10)	[2, 3, 4, 5, 10]
	(2, (3, 11))	[2, 3, 11]
	(2, (3, 4))	[2, 3, 4]
	(2, (5, 10))	[2, 3, 4, 5, 10]
	(2, (5, 6))	[2, 3, 4, 5, 6]
	(10, 4)	[10, 3, 4]
	(10, 6)	[10, 3, 4, 5, 6]
	(10, 8)	[10, 3, 4, 5, 6, 7, 8]
	(10, 10)	[10, 3, 4, 5, 10]
	(10, (3, 11))	[10, 3, 11]
	(10, (3, 4))	[10, 3, 4]
	(10, (5, 10))	[10, 3, 4, 5, 10]
	(10, (5, 6))	[10, 3, 4, 5, 6]

Variable	DU-pair	DU-path
j	(4, 6)	[4, 5, 6]
	(4, 8)	[4, 5, 6, 7, 8]
	(4, 9)	[4, 5, 6, 7, 8, 9]
	(4, (5, 10))	[4, 5, 10]
	(4, (5, 6))	[4, 5, 6]
	(9, 6)	[9, 5, 6]
	(9, 8)	[9, 5, 6, 7, 8]
	(9, 9)	[9, 5, 6, 7, 8, 9]
	(9, (5, 10))	[9, 5, 10]
	(9, (5, 6))	[9, 5, 6]
a	(6, (7, 9))	[6, 7, 9]
	(6, (7, 8))	[6, 7, 8]
b	(6, (7, 9))	[6, 7, 9]
	(6, (7, 8))	[6, 7, 8]

پرسش چهارم

(۱

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    // for (int i = 0; i < size; i++) {
    for (int i = 0; i < size - 1; i++) {
        rows[i] = 0;
        for (int j = 0; j < i; j++) {
            if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
                rows[i] = rows[j] + 1;
            }
        }
    }
}
```

(۲

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    // for (int i = 0; i < size; i++) {
    for (int i = 0; i < size + 1; i++) {
        rows[i] = 0;
        for (int j = 0; j < i; j++) {
            if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
                rows[i] = rows[j] + 1;
            }
        }
    }
}
```

(۳

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    // for (int i = 0; i < size; i++) {
    for (int i = 0; i < size; i += 2) {
        rows[i] = 0;
        for (int j = 0; j < i; j++) {
            if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
                rows[i] = rows[j] + 1;
            }
        }
    }
}
```

(۴

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    for (int i = 0; i < size; i++) {
        // rows[i] = 0;
        rows[i] = rows[i-1];
        for (int j = 0; j < i; j++) {
            if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
                rows[i] = rows[j] + 1;
            }
        }
    }
}
```

(۵)

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    for (int i = 0; i < size; i++) {
        rows[i] = 0;
        // for (int j = 0; j < i; j++) {
        for (int j = 0; j != i; j++) {
            if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
                rows[i] = rows[j] + 1;
            }
        }
    }
}
```

(۶)

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    for (int i = 0; i < size; i++) {
        rows[i] = 0;
        for (int j = 0; j < i; j++) {
            // if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
            if ((intervals[i].begin <= intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
                rows[i] = rows[j] + 1;
            }
        }
    }
}
```

(۷)

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    for (int i = 0; i < size; i++) {
        rows[i] = 0;
        for (int j = 0; j < i; j++) {
            // if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
            if ((intervals[i].begin < intervals[j].end) || (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
                rows[i] = rows[j] + 1;
            }
        }
    }
}
```

(۸)

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    for (int i = 0; i < size; i++) {
        rows[i] = 0;
        for (int j = 0; j < i; j++) {
            // if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
            if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] > rows[i]) {
                rows[i] = rows[j] + 1;
            }
        }
    }
}
```

۹)

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    for (int i = 0; i < size; i++) {
        rows[i] = 0;
        for (int j = 0; j < i; j++) {
            // if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
            if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] < rows[i]) {
                rows[i] = rows[j] + 1;
            }
        }
    }
}
```

۱۰)

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    for (int i = 0; i < size; i++) {
        rows[i] = 0;
        for (int j = 0; j < i; j++) {
            if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
                // rows[i] = rows[j] + 1;
                rows[i] = rows[j] - 1;
            }
        }
    }
}
```

۱۱)

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    for (int i = 0; i < size; i++) {
        rows[i] = 0;
        for (int j = 0; j < i; j++) {
            if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
                // rows[i] = rows[j] + 1;
                rows[i] = rows[i] + 1;
            }
        }
    }
}
```

۱۲)

```
struct interval {
    int begin;
    int end;
};

vector<int> layout(vector<interval> intervals, int size) {
    vector<int> rows(size);
    for (int i = 0; i < size; i++) {
        rows[i] = 0;
        for (int j = 0; j < i; j++) {
            if ((intervals[i].begin < intervals[j].end) && (intervals[i].end > intervals[j].begin) && rows[j] >= rows[i]) {
                rows[i] = rows[j] + 1;
                // immediate runtime failure if reached.
                bomb();
            }
        }
    }
```

پرسش پنجم

بله mutant شماره‌ی پنج redundant است.

پرسش ششم

- $intervals = \{\}, size = 0 \quad \rightarrow 2,$
- $intervals = \{\{8, 9\}\}, size = 1 \quad \rightarrow 1, 4$
- $intervals = \{\{8, 9\}, \{10, 11\}\}, size = 2 \quad \rightarrow 3, 7,$
- $intervals = \{\{8, 10\}, \{9, 11\}, \{8, 11\}\}, size = 3 \ \rightarrow 8, 9, 10, 11, 12$

بله هر چه تعداد mutant هایی که زنده میمانند بیشتر باشند، نشان دهنده‌ی ضعف بیشتر مجموعه تست کیس ها میباشد. همانطور که میدانیم mutant شماره پنج redundant است و mutant شماره شش توسط هیچ یک از تست های بالا کشته نمیشود. و تست زیر آن را میکشد:

$intervals = \{\{8, 9\}, \{9, 10\}\}, size = 2$