This is an open-book, open-notes, open-laptop, closed-network exam. Please do not post anything on Piazza until the solutions are put up on the class website.

You will have 2 hours to complete the exam, but the exam itself is designed to take no longer than 1 hour and 20 minutes. The examination has 6 problems worth 80 points. Please budget your time accordingly. Write your answers in the space provided on the exam. If you use additional scratch paper, please turn that in as well.

Your Name: ___________________ SUNet ID: ___________________

The following is a statement of the Stanford University Honor Code:

1. The Honor Code is an undertaking of the students, individually and collectively:
   
   (a) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

   (b) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

3. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

Signature: ___________________

<table>
<thead>
<tr>
<th>Problem</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>5</td>
<td>5</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>80</td>
</tr>
</tbody>
</table>
Problem 1. Garbage Collection [5 points]

For each of the metric below, is it true or false that generational garbage collection improves over the Baker’s algorithm. Explain your answer in a sentence or two.

1. Space reclaimed

2. Overall execution time

3. Space usage

4. Pause time

5. Data locality
Problem 2. Binary Decision Diagrams [5 points]

Consider the parity function $P_n$ over $n$ binary variables $x_1, x_2, ..., x_n$. $P_n$ outputs 0 if the number of variables with value 1 are even; $P_n$ outputs 1 otherwise.

1. Construct a reduced-ordered BDD for $P_3$ using the variable ordering $x_1 \geq x_2 \geq x_3$.

2. How many nodes (including the 0 and 1 terminal nodes) are there in the reduced-ordered BDD for $P_n$ with variable ordering $x_1 \geq x_2 \geq ... \geq x_n$?
Problem 3. Software Pipelining [20 points]

Consider the following data dependency graph. Next to each instruction are the resources used by that instruction.

1. What is the minimum initiation interval imposed by the precedence constraints?

2. What is the minimum initiation interval imposed by resource usage?
3. Run the heuristic scheduling algorithm described in class on this loop. Show both the modulo reservation table and the generated code (with nops if necessary) for a single iteration of the loop.

4. Is the initiation interval you found optimal? Explain.
Problem 4. Parallelization [20 points]

Consider the following program:

```java
for (int i = 1; i < N; i++) {
    for (int j = 0; j < i; j++) {
        X[i, j] = X[i+1, j] * Y[i, j+1] + X[0, j];  (1)
    }
    Y[i, i] = X[i, i];  (2)
}
```

1. Draw the iteration space for statements (1) and (2) on the same graph. Use arrows to mark data-dependencies between iterations. Label the iterations to indicate which statement they belong to.
2. Is there any communication-free parallelism that can be obtained using affine partitioning? If yes, what is the affine partitioning for statement (1), and for statement (2)?
3. Assuming \( p \) is the processor ID, what is the range of \( p \)? What is the SPMD code for each processor? You do not need to optimize the code generated; simply wrap statements with the right conditional statements.
Problem 5. Pointer Analysis [20 points]

Consider the following program.

```java
public class A {
    public A link;

    public void foo(A c) {
        c.link = new A(); // h1
    }

    public void bar(A d, A e) {
        d.link = e;
    }
}

public class B extends A {
    public void foo(A f) {
        f.bar(f, f);
    }

    public void bar(A g, A i) {
        g.link = new A(); // h2
        i.link = new B(); // h3
    }
}

public class Main {
    public static void main(String[] args) {
        A a = new A(); // h4
        a.foo(a);
        a = new B(); // h5
        a.foo(a);
    }
}
```
1. For each call site on lines 15, 27, 29, specify which methods are invoked when we apply a \textit{context-sensitive} and \textit{flow-sensitive} pointer analysis that computes the methods called on-the-fly. Distinguish clones with their IDs as subscripts. Each call site, and each method called, must be labeled by the clone ID. How many clones of A.foo will there be? How many clones of B.foo will there be?

2. Repeat question (1) above, but this time, use a \textit{context-insensitive} and \textit{flow-sensitive} pointer analysis. How many contexts for A.foo will there be? How many contexts for B.foo will there be?

3. Repeat question (1) again, but this time use a \textit{context-sensitive} and \textit{flow-insensitive} pointer analysis. How many clones of A.foo will there be? How many clones of B.foo will there be?
4. Perform a **context-insensitive** and **flow-insensitive** pointer analysis on the program. List the hpts tuples that are produced by the analysis.
Problem 6. Path-Sensitive Analysis with SMT [10 points]

Consider the following program, where data is a zero-indexed array with size N.

```cpp
void binary_search_step (int[] data, int N, int v, int &l, int &r) {
    if (0 <= l && l < r && r < N) {
        int mid = (l + r) / 2;
        if ( data[mid] <= v ) {
            l = mid;
        } else {
            r = mid - 1;
        }
    }
}
```

The SSA form of the program is given below:

```plaintext
phi0 = 0 <= l0 && l0 < r0 && r0 < N;
mid = (l0 + r0) / 2;
phi1 = data[mid] <= v;
l1 = mid;
r1 = mid - 1;
l2 = phi1 ? l1 : l0;
r2 = phi1 ? r0 : r1;
l3 = phi0 ? l0 : l2;
r3 = phi0 ? r0 : r2;
```

1. Property P1: The access to data on line 4 is always in bounds.

Write the formula for P1 using the notation of the SSA program above. (You do not need to use SMTLIB syntax when writing down the formulas. We’ll accept any format as long as it is readable.)
2. As the name implies, `binary_search_step` is intended to be used as a step in binary search. To ensure termination, this function is expected to satisfy the following property:

Property P2: Provided that the condition on line 2 holds, upon return of `binary_search_step`,

(a) $l_3 \leq r_3$,

(b) the range $[l_3, r_3]$ contains strictly fewer elements than the range $[l_0, r_0]$.

Write the formula for P2 using the notation of the SSA program above.

3. Suppose our SMT solver has a CHECKSAT function that accepts a given formula and returns if it can be satisfied (SAT), or not satisfied (UNSAT). How do you call CHECKSAT only once to determine whether the program satisfies both properties P1 and P2?

4. What is the ground truth for properties P1 and P2? For each property, answer "YES" if it is always true for any valid program input, and "NO" if otherwise. Assume there is no overflow behavior in the program. Justify your answers.