

T.11 · Weak compositions, loop invariants and flowchart tracing

Original classroom teaching; not an official form or score predictor

1. Scope and prerequisites

Scope and prerequisites

Undergraduate GRE preparation. Local objectives within the reviewed ETS scope; this is original teaching, not an official test or score predictor.

2. Scope and prerequisites

Scope and prerequisites

Prerequisites: Binomial coefficients, complements and assignment tracing.

3. Scope and prerequisites

Scope and prerequisites

Count indistinguishable allocations with nonnegative or positive constraints Trace a flowchart in statement order with correct reset behaviour Use a loop invariant and a progress measure to justify an algorithm

4. Scope and prerequisites

Scope and prerequisites

weak composition : An ordered allocation of a total into nonnegative integer parts.

5. Scope and prerequisites

Scope and prerequisites

loop invariant : A property preserved before and after each iteration of a loop.

6. Choose and justify a method

Choose and justify a method

The number of nonnegative integer solutions of $x_1 + \cdots + x_k = n$ is $C(n+k-1, k-1)$, for $n \geq 0$ and $k \geq 1$. Represent n identical stars separated into k named boxes by $k-1$ bars; adjacent bars and end bars allow zero entries. Boxes are distinguishable, objects are not. Counting distinct tokens would instead assign each token independently, giving k^n before other constraints. Decide the model from what the objects and recipients represent, not from a familiar-looking binomial coefficient.

7. Choose and justify a method

Choose and justify a method

For positive allocations $x_i \geq 1$, put $z_i = x_i - 1$; the remaining total is $n - k$, giving $C(n-1, k-1)$ when $n \geq k$. More general lower bounds $x_i \geq a_i$ are handled by subtracting each a_i . At least one zero is the complement of every entry positive: subtract $C(n-1, k-1)$ from the nonnegative total. Overlapping cases such as exactly two empty boxes need a different count; subtracting each empty-box case independently double-counts their intersections.

8. Choose and justify a method

Choose and justify a method

Trace a flowchart one executed statement at a time, recording variables in a table. Distinguish assignment from a mathematical equality test; the right side of an assignment uses the old value before replacement. Record which branch executes next and which variables reset. A new outer-loop candidate may reset an inner sum without resetting the candidate itself. For a process that adds 1,3,5,... to S from zero, after m additions $S=m^2$ and the next addend is $2m+1$.

9. Choose and justify a method

Choose and justify a method

A loop invariant is true before and after each iteration. The square-sum invariant holds initially at $m=0$ and is preserved because $m^2 + (2m+1) = (m+1)^2$. An invariant alone does not prove termination. To test whether an integer $N \geq 0$ is square by adding odd numbers until $S \geq N$, S grows without bound; equivalently m increases and must reach a bound such as N . At exit, equality means square and overshoot means nonsquare. A printed value must be reached on an actual execution path; a plausible numerical pattern alone is not a trace.

10. Worked reasoning

Worked reasoning

Distribute 11 identical tokens among 4 named boxes. Nonnegative allocations number $C(14,3)=364$; all-positive allocations number $C(10,3)=120$. Therefore 244 allocations have at least one empty box. For the odd-sum test with $N=10$, S progresses 0,1,4,9,16; it overshoots and rejects 10. For $N=9$ it exits at $S=9$ after three additions. The next addends are 1,3,5,7,9; sum and next-addend columns must not be confused.

Figure 11: Worked reasoning

Odd-sum loop: $S = m^2$, $B = 2m + 1$ before the next addition

Iterations m	Sum S	Next addend B
0	0	1
1	1	3
2	4	5
3	9	7
4	16	9

Check the labelled values and conditions.

12. Conditions and counterexamples

Conditions and counterexamples

Identical objects and distinct objects use different models. At least one empty box is not the same as exactly one. In a flowchart, preserve update order and reset only variables that the executed path actually resets.

13. Try independently

Try independently

Count nonnegative solutions of $a + b + c = 8$ with $a \geq 2$. How many also have $b, c \geq 1$? How many of the first set have at least one of b, c equal to zero?

14. Reasoning and check

Reasoning and check

Put $u = a - 2$. Then $u + b + c = 6$ has $\binom{8}{2} = 28$ nonnegative solutions. Requiring b, c positive and subtracting one from each gives total four in three nonnegative variables, counted by $\binom{6}{2} = 15$. The complement count is $28 - 15 = 13$. Directly $b=0$ gives seven, $c=0$ gives seven, and both zero is counted twice, giving $7 + 7 - 1 = 13$.

15. Try independently

Try independently

Trace this algorithm for $N=15$ and $N=16$: initialise $S=0$, $k=1$; while $S < N$, execute $S=S+k$, then $k=k+2$; report whether $S=N$. State an invariant, and justify termination for integer $N \geq 0$.

16. Reasoning and check

Reasoning and check

After m iterations, $S = m^2$ and $k = 2m + 1$. The trace pairs (S, k) are $(0, 1), (1, 3), (4, 5), (9, 7), (16, 9)$. Thus 15 is rejected by overshoot and 16 accepted. The invariant holds initially and is preserved since $m^2 + (2m + 1) = (m + 1)^2$. The sum grows without bound, so it eventually reaches or exceeds N ; for example $m = N$ suffices when N is positive. For $N = 0$ the loop is skipped and equality holds. An invariant alone would not establish termination.

Check yourself

Reconstruct the method without looking back

Trace this algorithm for $N=15$ and $N=16$: initialise $S=0$, $k=1$; while $S < N$, execute $S=S+k$, then $k=k+2$; report whether $S=N$. State an invariant, and justify termination for integer $N \geq 0$.