Total marks: 40

1. (a) flow across cut $= f(s,v_1)+f(v_2,v_1)+f(v_2,v_3)+f(v_4,v_3)+f(v_4,t) = 11+1+(-4)+7+4 = 19$ capacity of cut $= C(s,v_1)+C(v_2,v_1)+C(v_2,v_3)+C(v_4,v_3)+C(v_4,t) = 16+4+0+7+4 = 31$
   (b) min. cut is given by $S=\{s,v_1,v_2\}, T=\{v_3,v_4,t\}$
   (c) $C_f(u,v)+C_f(v,u) = (C(u,v)-f(u,v))+(C(v,u)-f(v,u))$ (by definition of C_f) $= C(u,v)+C(v,u)-f(u,v)-(-f(u,v))$ (by skew symmetry of f) $= C(u,v)+C(v,u)$
   (d) We show that f_p is a flow by proving that f_p satisfies the three properties of a flow:
      i. $f_p(u,v) \le C_f(p) \le C_f(u,v)$ for all (u,v) on p, and for all the other (u,v), $f_p(u,v) = 0 \le C_f(u,v)$.
      ii. Suppose (u,v) is on p. Then, $f_p(u,v) = C_f(p)$ and $f_p(v,u) = -C_f(p)$, so $f_p(u,v) = -f_p(v,u)$. For all other (u,v), $f_p(u,v) = 0 = f_p(v,u)$, so $f_p(u,v) = -f_p(v,u)$.
      iii. Suppose u is not on p. Then, (u,v) is assigned zero flow for any v, so the third property is satisfied trivially. If, on the other hand, $u \neq s,t$ is on the path, then it has an incoming flow from v (the previous vertex in the augmenting path) and an outgoing flow to w (the next vertex in path), which cancel out.

[5] 2. The corresponding decision problem is TSPD:

Instance: A distance matrix $D(i,j)$, $1 \le i,j \le n$, and a number B.

Question: Is there a tour of total distance B or less?

Here is a "guess-and-check" algorithm which shows that TSPD is in NP:

Guess: A tour $1, \pi(2), \pi(3), \dots, \pi(n), 1$, where $\{\pi(2), \pi(3), \dots, \pi(n)\}$ is a permutation of $\{2,3,\dots,n\}$.

Check: $D(1,\pi(2)) + D(\pi(2),\pi(3)) + \cdots + D(\pi(n-1),\pi(n)) \le B$.

We now sketch an algorithm showing how to efficiently reduce TSP to TSPD:

Let TSPD(D,B) be our "black box" procedure for solving TSPD. We use binary search on B to find the smallest B such that TSPD(D,B) = "YES" — let B_0 be the smallest such B. Now, we consider each edge (i,j) one at a time, and for each define the distance matrix D' as follows: $D' = D$ except that $D'(i,j) = B_0 + 1$. If TSPD(D', B_0) = "YES" then we know that the edge (i,j) is unnecessary and can be eliminated. Otherwise, (i,j) is necessary and must be kept.

not marked 3. Your TM should compare the first symbol of x with the first symbol of y, and then return the head to the first symbol of x, erase it, shift right, and continue. In general, symbols of y are erased one at a time, but only after the head returns to check the next symbol. In this way, the head is certain to hit a "guard" symbol in y, and will not run on forever if the next symbol of y is missing. The very first run comparing the first symbol uses different states than the later comparisons, because there is no guard symbol. This is a high level description of a TM that you shoud keep in mind when designing an actual TM with $\Gamma, \Sigma, Q, \delta$.

[5] 4. Let M' be the same as M except the states q_{accept} and q_{reject} are interchanged, i.e. if $\delta(q,s) = (q_{\text{accept}}, s, h)$, then $\delta'(q,s) = (q_{\text{reject}}, s, h)$, and same thing for q_{reject}.

[5] 5. In fact NSP $\in P$, and since $P \subseteq NP$, it follows that NSP $\in NP$. To see that NSP $\in P$, apply Kruskal's algorithm to G with edge cost function c to obtain a min cost ST $T_{\min}$. Then, an instance is a "YES" instance iff $c(T_{\min}) > B$.

[5] 6. (a) **Guess:** $S_0 \subseteq \{1,2,\dots,d\}$, such that $S_0 \neq \emptyset$, and **Check:** $\sum_{i \in S_0} w_i = 0$.

[5] (b) To show that PARTITION $\leq_p$ ZERO SUM, we show how to transform an instance $I = \langle d, w_1, w_2, \ldots, w_d \rangle$ of PARTITION to an instance I' of ZERO SUM in polytime, so that $I \in$ PARTITION iff $I' \in$ ZERO SUM:
Let $W = \sum_{i=1}^{d} w_i$. If W is odd, let $I' = \langle 1, 1 \rangle$ (a "NO" instance of ZERO SUM). If W is even, set $I' = \langle d+1, w_1, w_2, \ldots, w_d, -\frac{1}{2}W \rangle$.