technical

C5. Let k be a positive integer. The organising committee of a tennis tournament is to schedule the matches for $2k$ players so that every two players play once, each day exactly one match is played, and each player arrives to the tournament site the day of his first match, and departs the day of his last match. For every day a player is present on the tournament, the committee has to pay 1 coin to the hotel. The organisers want to design the schedule so as to minimise the total cost of all players' stays. Determine this minimum cost.

Answer: The required minimum is $k(4k^2+k-1)/2$.

Solution 1. Enumerate the days of the tournament $1, 2, \ldots, \binom{2k}{2}$. Let $b_1 \leqslant b_2 \leqslant \cdots \leqslant b_{2k}$ be the days the players arrive to the tournament, arranged in *nondecreasing* order; similarly, let $e_1 \geqslant \cdots \geqslant e_{2k}$ be the days they depart arranged in *nonincreasing* order (it may happen that a player arrives on day b_i and departs on day e_j, where $i \neq j$). If a player arrives on day b and departs on day e, then his stay cost is $e-b+1$. Therefore, the total stay cost is

$$\Sigma = \sum_{i=1}^{2k} e_i - \sum_{i=1}^{2k} b_i + n = \sum_{i=1}^{2k} (e_i - b_i + 1).$$

Bounding the total cost from below. To this end, estimate $e_{i+1} - b_{i+1} + 1$. Before day b_{i+1}, only i players were present, so at most $\binom{i}{2}$ matches could be played. Therefore, $b_{i+1} \leqslant \binom{i}{2} + 1$. Similarly, at most $\binom{i}{2}$ matches could be played after day e_{i+1}, so $e_i \geqslant \binom{2k}{2} - \binom{i}{2}$. Thus,

$$e_{i+1} - b_{i+1} + 1 \geqslant \binom{2k}{2} - 2\binom{i}{2} = k(2k-1) - i(i-1).$$

This lower bound can be improved for $i > k$: List the i players who arrived first, and the i players who departed last; at least $2i - 2k$ players appear in both lists. The matches between these players were counted twice, though the players in each pair have played only once. Therefore, if $i > k$, then

$$e_{i+1} - b_{i+1} + 1 \geqslant \binom{2k}{2} - 2\binom{i}{2} + \binom{2i-2k}{2} = (2k-i)^2.$$

An optimal tournament, We now describe a schedule in which the lower bounds above are all achieved simultaneously. Split players into two groups X and Y, each of cardinality k. Next, partition the schedule into three parts. During the first part, the players from X arrive one by one, and each newly arrived player immediately plays with everyone already present. During the third part (after all players from X have already departed) the players from Y depart one by one, each playing with everyone still present just before departing.

In the middle part, everyone from X should play with everyone from Y. Let $S_1, S_2, \ldots, S_k$ be the players in X, and let $T_1, T_2, \ldots, T_k$ be the players in Y. Let $T_1, T_2, \ldots, T_k$ arrive in this order; after T_j arrives, he immediately plays with all the S_i, $i > j$. Afterwards, players S_k, $S_{k-1}, \ldots, S_1$ depart in this order; each S_i plays with all the T_j, $i \leqslant j$, just before his departure, and S_k departs the day T_k arrives. For $0 \leqslant s \leqslant k-1$, the number of matches played between T_{k-s}'s arrival and S_{k-s}'s departure is

$$\sum_{j=k-s}^{k-1} (k-j) + 1 + \sum_{j=k-s}^{k-1} (k-j+1) = \frac{1}{2}s(s+1) + 1 + \frac{1}{2}s(s+3) = (s+1)^2.$$

Thus, if $i > k$, then the number of matches that have been played between T_{i-k+1}'s arrival, which is b_{i+1}, and S_{i-k+1}'s departure, which is e_{i+1}, is $(2k-i)^2$; that is, $e_{i+1} - b_{i+1} + 1 = (2k-i)^2$, showing the second lower bound achieved for all $i > k$.

If $i \leqslant k$, then the matches between the i players present before b_{i+1} all fall in the first part of the schedule, so there are $\binom{i}{2}$ such, and $b_{i+1} = \binom{i}{2} + 1$. Similarly, after e_{i+1}, there are i players left, all $\binom{i}{2}$ matches now fall in the third part of the schedule, and $e_{i+1} = \binom{2k}{2} - \binom{i}{2}$. The first lower bound is therefore also achieved for all $i \leqslant k$.

Consequently, all lower bounds are achieved simultaneously, and the schedule is indeed optimal.

Evaluation. Finally, evaluate the total cost for the optimal schedule:

$$\begin{aligned}\Sigma &= \sum_{i=0}^{k}\big(k(2k-1) - i(i-1)\big) + \sum_{i=k+1}^{2k-1}(2k-i)^2 = (k+1)k(2k-1) - \sum_{i=0}^{k} i(i-1) + \sum_{j=1}^{k-1} j^2 \\ &= k(k+1)(2k-1) - k^2 + \frac{1}{2}k(k+1) = \frac{1}{2}k(4k^2+k-1).\end{aligned}$$

Solution 2. Consider any tournament schedule. Label players $P_1, P_2, \ldots, P_{2k}$ in order of their arrival, and label them again $Q_{2k}, Q_{2k-1}, \ldots, Q_1$ in order of their departure, to define a permutation $a_1, a_2, \ldots, a_{2k}$ of $1, 2, \ldots, 2k$ by $P_i = Q_{a_i}$.

We first describe an optimal tournament for any given permutation $a_1, a_2, \ldots, a_{2k}$ of the indices $1, 2, \ldots, 2k$. Next, we find an optimal permutation and an optimal tournament.

Optimisation for a fixed $a_1, \ldots, a_{2k}$. We say that the *cost* of the match between P_i and P_j is the number of players present at the tournament when this match is played. Clearly, the Committee pays for each day the cost of the match of that day. Hence, we are to minimise the total cost of all matches.

Notice that Q_{2k}'s departure does not precede P_{2k}'s arrival. Hence, the number of players at the tournament monotonically increases (non-strictly) until it reaches $2k$, and then monotonically decreases (non-strictly). So, the best time to schedule the match between P_i and P_j is either when $P_{\max(i,j)}$ arrives, or when $Q_{\max(a_i,a_j)}$ departs, in which case the cost is $\min\big(\max(i,j), \max(a_i,a_j)\big)$.

Conversely, assuming that $i > j$, if this match is scheduled between the arrivals of P_i and P_{i+1}, then its cost will be exactly $i = \max(i,j)$. Similarly, one can make it cost $\max(a_i,a_j)$. Obviously, these conditions can all be simultaneously satisfied, so the minimal cost for a fixed sequence $a_1, a_2, \ldots, a_{2k}$ is

$$\Sigma(a_1, \ldots, a_{2k}) = \sum_{1 \leqslant i < j \leqslant 2k} \min\big(\max(i,j), \max(a_i,a_j)\big). \tag{1}$$

Optimising the sequence (a_i). Optimisation hinges on the lemma below.

Lemma. If $a \leqslant b$ and $c \leqslant d$, then

$$\begin{aligned}&\min\big(\max(a,x), \max(c,y)\big) + \min\big(\max(b,x), \max(d,y)\big) \\ &\qquad\qquad \geqslant \min\big(\max(a,x), \max(d,y)\big) + \min\big(\max(b,x), \max(c,y)\big).\end{aligned}$$

Proof. Write $a' = \max(a,x) \leqslant \max(b,x) = b'$ and $c' = \max(c,y) \leqslant \max(d,y) = d'$ and check that $\min(a',c') + \min(b',d') \geqslant \min(a',d') + \min(b',c')$. □

Consider a permutation $a_1, a_2, \ldots, a_{2k}$ such that $a_i < a_j$ for some $i < j$. Swapping a_i and a_j does not change the (i,j)th summand in (1), and for $\ell \notin \{i,j\}$ the sum of the (i,ℓ)th and the (j,ℓ)th summands does not increase by the Lemma. Hence the optimal value does not increase, but the number of disorders in the permutation increases. This process stops when

$a_i = 2k + 1 - i$ for all i, so the required minimum is

$$S(2k, 2k-1, \dots, 1) = \sum_{1 \leqslant i < j \leqslant 2k} \min\bigl(\max(i, j), \max(2k+1-i, 2k+1-j)\bigr)$$
$$= \sum_{1 \leqslant i < j \leqslant 2k} \min(j, 2k+1-i).$$

The latter sum is fairly tractable and yields the stated result; we omit the details.

Comment. If the number of players is odd, say, $2k-1$, the required minimum is $k(k-1)(4k-1)/2$. In this case, $|X| = k$, $|Y| = k-1$, the argument goes along the same lines, but some additional technicalities are to be taken care of.