UCB

In [8]:

```
H = 10000

actions_Q = [0, 0, 0]
actions_U = [0, 0, 0] #correspond aux U_t(a)
actions_N = [1, 1, 1] #initialisation à 1 pour éviter une division par 0

actions_compteurs = [[0], [0], [0]]
```

In [9]:

```
for t in range(1, H):
    #calcul de U_t(a) pour toutes les actions a
    for a in range(len(actions_q)):
        actions_U[a] = np.sqrt(2 * np.log(t) / actions_N[a])

    a = indice_max(ajouter_listes(actions_Q, actions_U))

    actions_N[a] += 1
    r = generer_reponse(actions_q, a)

    actions_Q[a] = actions_Q[a] + (1/actions_N[a]) * (r - actions_Q[a])

    actions_compteurs[a].append(actions_compteurs[a][-1] + 1)
    for b in range(len(actions_q)):
        if b != a:
            actions_compteurs[b].append(actions_compteurs[b][-1])
```

In [10]:

```
actions_Q
```

Out[10]:

```
[0.8990328253223917, 0.0, 0.8745644599303144]
```

In [11]:

```
plt.plot(actions_compteurs[0], label='0.9')
plt.plot(actions_compteurs[1], label='0.1')
plt.plot(actions_compteurs[2], label='0.8')

plt.xlabel('temps')
plt.ylabel('actions cumulées')
plt.legend()

plt.savefig('ucb.png', dpi=300)
```